www.sapientia.uji.es | 15



Informática Básica. Conceptos básicos de informática

Juan Carlos Amengual Argudo

Vicente Ramón Tomás López

Informática Básica. Conceptos básicos de informática

Juan Carlos Amengual Argudo
Vicente Ramón Tomás López



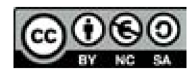
Departamento de Lenguajes y Sistemas Informáticos
Departamento de Ingeniería y Ciencia
de los Computadores

■ Codi d'assignatura IX01

Edita: Publicacions de la Universitat Jaume I. Servei de Comunicació i Publicacions
Campus del Riu Sec. Edifici Rectorat i Serveis Centrals. 12071 Castelló de la Plana
http://www.tenda.uji.es e-mail: publicacions@uji.es

Col·lecció Sapientia, 15
www.sapientia.uji.es

ISBN: 978-84-692-3984-1



PARTE I
CONCEPTOS DE INFORMÁTICA

ÍNDICE GENERAL

CAPÍTULO 1

PRESENTACIÓN

1.1. LA ASIGNATURA INFORMÁTICA BÁSICA

La asignatura Informática Básica es una asignatura de libre configuración, aunque recomendada para los nuevos estudiantes de las titulaciones de informática de la Universitat Jaume I (UJI):

- Ingeniería Informática,
- Ingeniería Técnica en Informática de Gestión e
- Ingeniería Técnica en Informática de Sistemas.

La asignatura está recomendada *muy especialmente para aquéllos que no han estudiado informática previamente*. No es posible matricularse de esta asignatura si tienes aprobada alguna asignatura de programación o estructura de ordenadores (de “informática” para que nos entendamos). Consta de $2{,}5$ créditos de teoría y de 2 créditos de prácticas y se imparte de forma intensiva durante las ocho primeras semanas del semestre. Las prácticas de la asignatura comenzarán la segunda semana del curso y abarcarán un total de siete semanas (concluyendo al mismo tiempo que la teoría).

En esta asignatura se introducirán conceptos básicos sobre el funcionamiento y el manejo de ordenadores, prestando especial atención a los sistemas operativos, sus entornos gráficos de trabajo, la red Internet y la seguridad electrónica. Cabe señalar que, al tratarse de una asignatura básica, no se asume ningún tipo de conocimiento previo por parte del alumno.

Se tratarán los dos componentes principales de un ordenador (*hardware* y *software*), así como las principales funciones del sistema operativo. También se introducirá al alumno en el manejo y configuración de los entornos gráficos de trabajo, y se explicarán las operaciones básicas para el manejo del sistema de ficheros. Para ello, se verán ejemplos de aplicación con dos sistemas operativos diferentes: Windows y Linux. En lo que se refiere a Internet, se introducirá al alumno en el manejo de los servicios más utilizados de entre los ofrecidos por las redes de ordenadores: correo

electrónico, navegación, transferencia de ficheros, conexión remota... Finalmente, se explicarán conceptos de seguridad electrónica para la transmisión de la información y se hablará también de copias de seguridad y de compresión de la información.

Los anteriores conceptos, presentados en las clases de teoría, podrán ser practicados por los alumnos en las correspondientes sesiones de trabajo en laboratorio.

En cuanto a la terminología empleada en esta colección docente, quisiéramos apuntar que, si bien es cierto que en los últimos años se está intentando introducir nuevos vocablos en castellano y catalán, hemos optado por presentar los términos en su versión original en inglés tradicionalmente aceptada. Es el caso, particularmente, de las palabras *hardware* y *software*, donde cada vez es más frecuente encontrarse en su lugar los equivalentes *maquinario* (castellano) o *maquinari* (catalán) y *programario* (castellano) o *programari* (catalán), respectivamente.

1.1.1. Objetivos

El objetivo principal de la asignatura consiste en que el alumno se familiarice con los conceptos básicos de la informática y con el uso de determinadas herramientas que le serán de mucha utilidad en el posterior transcurso de la carrera.

Desde este punto de vista, esta asignatura pretende paliar muchas de las carencias que, en cuanto a conocimientos básicos de informática, presentan los alumnos de nuevo ingreso en la Universidad. Por otra parte, contribuye a descargar los temarios de otras asignaturas de informática más específicas y menos apropiadas para la enseñanza de estos conceptos básicos.

En la práctica, esto se traduce en un temario que se ha elaborado con el fin de conseguir los siguientes objetivos concretos:

1. Introducir conceptos básicos sobre el funcionamiento y uso de los ordenadores. En particular, sobre:
   a) Sistemas operativos y entornos de trabajo para el usuario.
   b) Codificación de la información en los ordenadores.
   c) Hardware (la "circuitería" del ordenador).
   d) Redes de computadores e Internet.
   e) Seguridad y protección de la información.
   f) Software, esto es aplicaciones o programas informáticos.
2. Presentar y familiarizar al estudiante con algunos entornos de trabajo:
   a) Microsoft Windows XP.
   b) El intérprete de órdenes de Unix.
   c) Entorno KDE, sobre GNU Linux.
3. Introducir el uso de algunas aplicaciones básicas:

a) Un procesador de textos: Writer del paquete ofimático OpenOffice[1].

b) Un editor de texto: XEmacs.

c) Navegadores de Internet: Mozilla Firefox y Konqueror.

d) Un cliente de correo electrónico o "pasarela" vía web (*horde* e *imp*).

e) Utilidades para transferencia de ficheros (FTP y SSH/SCP usando clientes gráficos y `scp` y `ftp` desde la línea de órdenes).

f) Terminal para conexiones remotas con cifrado de datos (`ssh`).

g) ...

1.1.2. Teoría

En las clases de teoría se presentan los conceptos básicos expuestos con anterioridad. Puesto que muchos de los conocimientos impartidos en las clases de teoría serán utilizados durante la realización de las sesiones de prácticas, resulta conveniente llevar la teoría "al día" para poder efectuar con éxito las prácticas. En el Cuadro 1.1 se expone el temario de teoría de la asignatura.

Tema	**Título**	**Sesiones** (aprox.)
0	Presentación	1
1	Sistemas operativos y entornos de usuario	2
2	Codificación de la información	2–3
3	Hardware	3–4
4	Redes de computadores: Internet	2–3
5	El intérprete de órdenes Unix	1–2
6	Seguridad y protección de la información	1–2

Cuadro 1.1. Temario de teoría de la asignatura Informática Básica

En la página web de la asignatura encontraréis copias reducidas[2] de las transparencias usadas en clase, así como cuadernos de apuntes[3], uno por cada tema, que permiten profundizar en las materias presentadas y ampliar conocimientos. También podemos dejar este material en reprografía si así lo solicitáis.

Es importante que asistáis a clase con, al menos, la copia reducida de las transparencias. De ese modo tendréis que tomar menos apuntes y podréis prestar más atención a las explicaciones. Las transparencias estarán disponibles en reprografía y en la página web de la asignatura antes de comenzar la primera clase de la materia correspondiente.

1.1.3. Prácticas

Durante las sesiones de prácticas se desarrollarán algunos de los conceptos explicados durante las clases de teoría. Todas las sesiones de prácticas se realizan en aulas

[1]OpenOffice tiene dos ventajas: es software libre y es multiplataforma. Esto último quiere decir que tenemos versiones tanto para Windows como para Linux, lo cual es muy importante.

[2]A ocho transparencias por página.

[3]Como éste.

informáticas dotadas de ordenadores personales (PC) equipados con sistemas operativos Windows XP y GNU Linux (distribución CENTOS, derivada de Red Hat Linux).

Para las sesiones prácticas se dispondrá tanto de un boletín, que indica específicamente los ejercicios a realizar, como de un cuadernillo, con material de apoyo necesario para la correcta realización del boletín de prácticas. El calendario previsto de prácticas es el siguiente:

Práctica	**Materia**	**Sesiones** (aprox.)
1	El entorno de trabajo Microsoft Windows XP	1
2	El entorno de trabajo KDE sobre GNU Linux	1–2
3	El procesador de textos Writer (OpenOffice)	2–3
4	El editor de textos XEmacs	2–3
5	Internet (navegación, búsquedas, correo-e, FTP, . . .)	3–4
6	El intérprete de órdenes Unix	2

Cuadro 1.2. Calendario de prácticas de la asignatura Informática Básica

En las prácticas, el profesor ofrecerá una breve introducción a cada materia. Trabajaréis sobre guiones (los boletines de prácticas) que marcarán una serie de objetivos a alcanzar. El guión puede complementarse con material adicional (ampliación de conceptos, manuales, archivos de trabajo, etc.).

El material estará disponible en la página web de la asignatura, al menos, el día antes de la sesión práctica correspondiente. También podemos dejarlo en reprografía si queréis. Es importante que leáis el guión y conozcáis bien el material de apoyo adicional[4] antes de empezar cada sesión práctica. De lo contrario, es posible que no dispongáis de tiempo suficiente.

1.1.4. Evaluación

La evaluación de la asignatura se realizará teniendo en cuenta dos posibles itinerarios alternativos y mutuamente excluyentes:

Itinerario A**:** Este itinerario conlleva una evaluación continua del *trabajo semanal* realizado en la asignatura y, por tanto, requiere *asistencia obligatoria* a clase, con el fin de que puedan seguirse adecuadamente todas las tareas programadas.

Itinerario B**:** Podríamos considerarlo como el sistema “tradicional”, ya que no se requiere asistencia a clase y el alumno fía la nota al resultado obtenido en el examen final.

En ambos casos habrá que realizar una prueba práctica en un aula de informática. Se trata del examen final al concluir las clases. Sin embargo, hay que hacer constar que el itinerario B requerirá una *prueba teórica adicional*. Por su parte, el itinerario A requiere la entrega de algunos trabajos y la realización de ciertas actividades durante el curso. Con ambos itinerarios se guardan los resultados de las *partes aprobadas*, de cara

[4]Con leérselo al menos una vez sería suficiente.

a una hipotética segunda convocatoria en septiembre (recuperación). El profesorado de la asignatura recomienda encarecidamente que el alumno siga el itinerario A. También hacemos constar que este itinerario requiere que los alumnos formen grupos de trabajo, ya que hay ciertas actividades colectivas. Se informará oportunamente al respecto en las clases de teoría, donde se firmará un contrato de aprendizaje.

La prueba de teoría (sólo itinerario B) constará de una serie de cuestiones breves. La prueba práctica constará de una serie de ejercicios que se realizarán con ordenador. En la prueba práctica (solamente) se podrá consultar cualquier material de apoyo que se considere oportuno (apuntes de clase, transparencias, libros. . .).

La calificación definitiva de la asignatura se obtendrá mediante una media ponderada de la nota obtenida en cada una de las pruebas (itinerario B) o bien de la nota obtenida en la prueba práctica y las actividades y trabajos realizados durante el curso (itinerario A). En resumen:

- Itinerario A: evaluación continua, asistencia obligatoria, con contrato de aprendizaje:
  - Examen de prácticas: $55\,\%$ de la nota final; se requiere una calificación mínima de 4 para poder aprobar la asignatura.
  - Actividades y trabajos: $45\,\%$ de la nota final; se requiere una calificación mínima de 5 para poder aprobar la asignatura.
- Itinerario B: no se requiere asistencia a clase y no hay contrato de aprendizaje:
  - Examen de teoría: $40\,\%$ de la nota final; se requiere una calificación mínima de 4 para poder aprobar la asignatura.
  - Examen de prácticas: $60\,\%$ de la nota final; se requiere una calificación mínima de 4 para poder aprobar la asignatura.

El examen de la asignatura se celebrará *fuera del periodo ordinario* de exámenes de enero/febrero, concretamente el ***primer día que tengamos aulas disponibles tras la semana en que concluyen las clases***. El motivo de realizarlo así obedece, fundamentalmente, al hecho de intentar hacer el examen lo antes posible, cuando todavía tenéis "frescos" los conceptos vistos en clases de teoría y trabajados en prácticas.

1.1.5. Bibliografía

El material bibliográfico que podéis consultar, tanto para repasar los conceptos explicados en clase, como para ampliar información al respecto, es el siguiente:

- Apuntes y transparencias de la asignatura editados en reprografía (también accesibles vía web). Este material se considera de lectura "obligada".
- *Introducción a la informática: hardware, software y teleinformática*, de Miguel Ángel Sánchez Vidales, Publicaciones Universidad Pontificia de Salamanca (2001).

- *Trabajar con la shell*, capítulo 10 del manual de configuración de SuSE LINUX 7.3.
- En la página web de la asignatura podréis encontrar enlaces a manuales on-line, cursos, documentos, artículos, etc., donde podréis ampliar información de varios de los temas abordados en la asignatura.

Además de ésta, es posible que en algunos temas (capítulos) se incorpore bibliografía específica utilizada en la elaboración del material o bien para ampliar algunos de los conceptos explicados.

1.1.6. Página web

La asignatura tiene su propia página web en el *Aula Virtual* de la UJI con información sobre la misma, apuntes, boletines de prácticas, enlaces a páginas con material relacionado, etc. Podrás acceder a ella en la dirección:

`http://aulavirtual.uji.es`

1.2. INTRODUCCIÓN A LOS CONTENIDOS DE LA ASIGNATURA

El término *informática* hace referencia al conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de ordenadores. De esta definición podemos destacar dos conceptos:

Tratamiento de la información: hace referencia a la aplicación sistemática de uno o varios programas sobre un conjunto de datos, con el fin de utilizar la información que contienen. Esto es lo que también se conoce como *proceso de datos*. Como veremos en el *capítulo 3*, la información, esto es, los datos que procesa el ordenador, debe ser adecuadamente representada en el mismo para poder ser tratada → *codificación de la información*.

Ordenador: es un dispositivo (máquina) electrónico(a) dotado(a) de una memoria de gran capacidad y de métodos de tratamiento de la información, capaz de resolver problemas aritméticos y lógicos gracias a la utilización automática de *programas* registrados en ella. Por tanto, no sólo realiza, básicamente, cálculos aritméticos y operaciones lógicas, sino que es capaz de ejecutar programas con instrucciones más complejas y procesar datos, imágenes, sonido, etc.

Gracias a la enorme velocidad a la que es capaz de realizar las operaciones básicas, un ordenador es capaz de recibir todo tipo de información, procesarla y, una vez procesada, emitirla al usuario ya "digerida" para su interpretación.

Técnicamente hablando, el ordenador es un conjunto de circuitos electrónicos comprimidos en pastillas de silicio, llamadas *chips*, siendo su función fundamental la de enviar señales electromagnéticas de un dispositivo a otro.

Hardware: conjunto *físico* de todos los dispositivos y elementos internos y externos del ordenador, esto es los dispositivos donde se realizan los cálculos y acciones del sistema.

Software: conjunto de datos, programas, informaciones. . . El software son todas las instrucciones y datos que se ejecutan/almacenan dentro del ordenador. No tiene "existencia material" (aunque se presente sobre un soporte físico como disquete, CD-ROM, disco duro, etc.). El software se puede copiar fácilmente y con total fiabilidad, lo cual plantea problemas de copyright, es decir, derechos de autor. Todo lo que maneja el ordenador es software, aunque puede repercutir en el hardware, por ejemplo si usamos un programa para configurar ciertos parámetros de la tarjeta de sonido.

Estableciendo una analogía, en un libro, el papel, las tapas y la tinta constituyen el hardware, las palabras (el texto escrito, las ideas que en él se vierten) son el software. Echándole un poco de ironía al asunto, también podríamos decir que:

- Hardware: lo que puedes partir con un hacha.
- Software: aquello que sólo puedes maldecir.

1.2.1. Hardware

El tipo de procesador y la clase de dispositivos con los que se conecta, la forma en que se conectan entre sí los distintos elementos y la manera en que está hecha la circuitería y su modo de funcionamiento determinan la *arquitectura* de la máquina, esto es el *tipo* o la clase de ordenador. Cada ordenador se diseña de acuerdo con una cierta arquitectura.



Figura 1.1. Un ordenador Personal (PC o x86)

Los dispositivos son específicos para determinadas arquitecturas (por la forma de conexión, tipos de buses, etc.). Más aún, los programas se diseñan y preparan para ser ejecutados en arquitecturas concretas. Por tanto, si tenemos un tipo de ordenador determinado y queremos ejecutar un programa concreto, deberemos buscar la versión adecuada para nuestro ordenador (arquitectura).

Existen varias arquitecturas: m68k (Mac), PowerPC, Sparc (SUN), etc. Vamos a centrarnos en la arquitectura más popular y extendida, no sólo por razones históricas, sino también por su notable rendimiento, la buena relación calidad/precio y fuerte grado de desarrollo que ha experimentado en los últimos años: la arquitectura PC (de *P*ersonal *C*omputer), también conocida como *x86*. Los principales elementos hardware que podemos encontrar en un PC son:

- La **CPU** (del inglés *C*entral *P*rocessing *U*nit, Unidad Central de Proceso en castellano) o **microprocesador**. Es la parte fundamental de un ordenador, el verdadero "cerebro electrónico" que gobierna el funcionamiento de la máquina.

  Todo lo que le rodea y que se conecta bien de forma directa, bien de forma indirecta, no son más que dispositivos, mediante los cuales el microprocesador interactúa con los usuarios. La CPU se compone, a su vez, de la Unidad Aritmético-Lógica (UAL) y de la Unidad de Control (UC). Esta unidad trabaja en base a un reloj maestro que coordina la ejecución de todas las operaciones que realiza el microprocesador. La unidad fundamental de trabajo de este reloj es la cantidad de instrucciones que el microprocesador puede ejecutar en un segundo. Esto viene determinado por la frecuencia del reloj, la cual se mide hoy en día en GHz (antes en MHz), esto es, el número de ciclos de reloj por segundo. Un procesador a 2,4 GHz puede realizar 2.400 millones de ciclos por segundo. Sin embargo, debes tener presente que *muchas de las instrucciones del procesador requieren más de un ciclo de reloj para completarse*.

- La **memoria principal**, donde se guardan los programas en ejecución y los datos que éstos requieren y/o manipulan. Para la memoria principal del ordenador (la de "funcionamiento") se utiliza memoria RAM (*Random Access Memory*). Se puede leer y también se puede escribir en ella. Tiene la característica de ser volátil, esto es, que sólo funciona mientras esté encendido el ordenador (necesita alimentación de la corriente eléctrica).

  En ella se almacenan tanto las instrucciones que necesita ejecutar el microprocesador como los datos que introducimos y deseamos procesar, así como los resultados obtenidos. Su unidad de almacenamiento es el *byte*, el espacio que se necesita para almacenar un carácter. Hoy en día se mide en Gigabytes (GB). Para que te hagas una idea, 1 GB son, aproximadamente, mil millones de bytes (2^{30} para ser exactos).

- El **bus**. Los buses son las pistas de comunicación que conectan entre sí diversos elementos. Por tanto, sirven para enviar información entre las partes del computador.

¿Sabías que…?

Los tipos de buses que podemos encontrar habitualmente en los PC actuales son el PCI (y su versión más moderna PCI-Express) y los nuevos estándares: AGP para tarjetas de vídeo (o tarjetas gráficas) y el Universal Serial Bus (USB) para conexión con componentes externos al PC.

Hoy en día también encontramos placas equipadas con FireWire (nombre "popular" de la conexión IEEE1394). Aunque ya no es tan habitual, es posible que también podamos encontrar un bus EISA.

AGP y PCI son los dos tipos de ranuras o *slots* compatibles con las tarjetas actuales (AGP *exclusivamente* para tarjetas gráficas).

continúa en la página siguiente …

. . . viene de la página anterior

El bus EISA se suele poner por compatibilidad (lo usan tarjetas más antiguas, aunque no las que se fabrican actualmente).

- Las **tarjetas de vídeo y sonido**. La primera (también conocida como tarjeta gráfica) es la responsable de generar las imágenes que se visualizan en el monitor. La tarjeta gráfica es la que se encarga de procesar la información que el procesador le envía y, a su vez, de enviarla al monitor, para que éste muestre lo que ocurre en el computador. Mantiene la imagen que se ha de mostrar en pantalla almacenada en su propia memoria. La segunda se ocupa de procesar todo el audio del sistema (multimedia).

¿Sabías que. . . ?

Hoy en día las tarjetas gráficas son para ranuras AGP y PCI-Express. Las tarjetas generalmente tienen algún tipo de aceleración (procesador específico). No se deben confundir con las tarjetas sintonizadoras de televisión (para ver televisión en el ordenador) o las capturadoras de vídeo, a las cuales se les puede conectar reproductores de VHS.

- La **placa base**, *motherboard* en inglés. Es la circuitería a la cual se conectan los anteriores elementos. En ella se encuentran las conexiones básicas para todos los componentes del ordenador, los cuales giran en torno al microprocesador. Permite o no el futuro crecimiento de la capacidad de cualquier ordenador, en función del tipo de buses, número de *ranuras* para insertar tarjetas, velocidad del bus, etc.
- Los **periféricos**, los cuales conectan el ordenador con el "mundo exterior". Los hay:
  - de **entrada**. Para introducir datos en nuestro ordenador: teclado, ratón, joystick, micrófono, escáner, lector de CD-ROM. . .
  - de **salida**. Para que el ordenador muestre los resultados del proceso de datos efectuado o simplemente se extraigan datos de él: monitor, impresora, altavoces. . .
  - de **entrada/salida**. Permiten tanto introducir datos como extraerlos: discos duros, regrabadora de CD-RW (regrabable), módems. . .
- La **fuente de alimentación**. Es como una especie de *transformador*, ya que distribuye la energía eléctrica entre los distintos elementos, haciendo que funcionen al voltaje adecuado.
- La **caja** también es un elemento importante, ya que sirve de armazón sobre el que se montan los elementos internos del ordenador.

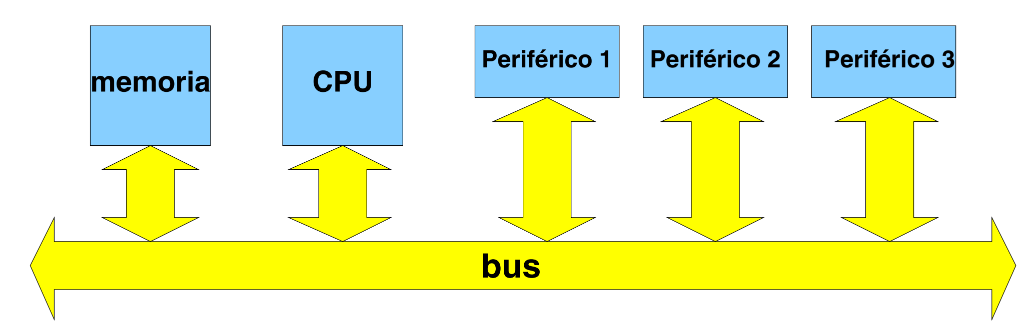
Ten en cuenta que…

La caja puede ser un elemento importante aunque no lo parezca. Debe tener ciertas características que la hagan funcional:

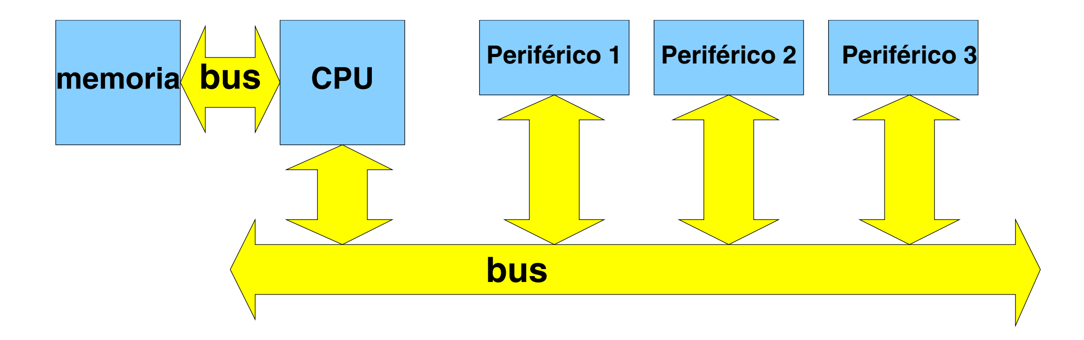
- Solidez.
- Número de "huecos" suficientes para las unidades que se desean colocar.
- Una buena fuente de alimentación y al voltaje adecuado. Un voltaje incorrecto puede dañar otras piezas.
- Fácil acceso para la instalación de dispositivos.
- Configuración compatible con la placa base que se va a instalar.

Con respecto a esto último, existen unas normas estandarizadas que regulan las características de las torres. Los procesadores antiguos (486, 586, Pentium, K5, etc.) utilizaban generalmente el estándar AT. Hoy en día el estándar es el ATX. Se debe verificar que la torre sea compatible con la placa base. En las características de la placa se puede leer de qué tipo es.

Un posible esquema general de funcionamiento de los dispositivos que conforman el ordenador:



Ahora puedes ver una disposición alternativa, más acorde con la realidad. Aquí tenemos un bus exclusivo para la comunicación entre la memoria principal y la CPU.



1.2.2. Software

El ordenador, realmente, sería una máquina sin utilidad si no fuera por los programas que puede ejecutar y la información que a éstos les es posible manipular; es decir el *software*. Por ejemplo, ¿qué podríamos hacer con un ordenador que no tuviese instalado un sistema operativo[5]?

Un *programa* es la descripción de un *algoritmo* en un *lenguaje de programación* que el ordenador es capaz de interpretar. Desde el punto de vista del sistema operativo, un programa en ejecución y los datos que éste maneja constituyen lo que se llama un *proceso*. Así pues, hablaremos más de procesos que de programas en ejecución.

Ahora bien, ¿qué es un algoritmo? Un algoritmo es una receta para resolver un problema. Otra definición más "formal": un algoritmo es una secuencia de operaciones que permiten obtener un resultado a partir de una serie de datos de entrada.

Otra cuestión. ¿Qué es un lenguaje de programación? Los lenguajes de programación son lenguajes artificiales que definen mediante unas reglas muy precisas y rigurosas:

- qué palabras se pueden usar, esto es, el *léxico* del lenguaje.
- qué frases se pueden formar con las palabras, esto es, la *sintaxis* o reglas sintácticas que hay que aplicar en el lenguaje a la hora de formar frases válidas.
- cuál es el significado (*semántica*) de las frases válidas (sintácticamente correctas) que es posible construir.

Así pues, los lenguajes de programación permiten escribir programas de modo que puedan ser entendidos por un ordenador. Los programas, al estar escritos usando un lenguaje de programación, necesitan traducirse al *lenguaje máquina*. Esto es, básicamente, una descripción binaria de las instrucciones elementales que ha de ejecutar el microprocesador. Todos los "micros" tienen un conjunto o juego de instrucciones básicas predeterminado.

El programa que realiza esta traducción se denomina *compilador*. Los fabricantes de compiladores conocen muy bien el juego de instrucciones básico del procesador usado en la arquitectura para la cual construyen el compilador. Para que un programa pueda funcionar en un ordenador debe ser compilado previamente.

Algunas cosas más sobre algoritmos

En realidad, la definición más "formal" posible sería ésta: un algoritmo es un conjunto de instrucciones bien definidas que nos permite resolver un determinado problema en tiempo finito. Pero seguramente entenderás mejor las anteriores.

Las características que ha de poseer todo "buen" algoritmo para ser considerado como tal son:

continúa en la página siguiente ...

[5]El sistema operativo es un programa, al fin y al cabo software, que se encarga de controlar todos los recursos y proporcionar el soporte que necesitan los programas de usuario.

. . . viene de la página anterior

- **Finitud**: debe terminar en un tiempo *finito* (aunque puede ser muy grande).
- **Buena definición**: los pasos del algoritmo deben estar bien definidos.
- **Entrada**: el algoritmo recibe cero o más datos de entrada.
- **Salida**: el algoritmo produce una o más salidas.
- **Factibilidad**: cada paso debe poder llevarse a cabo.

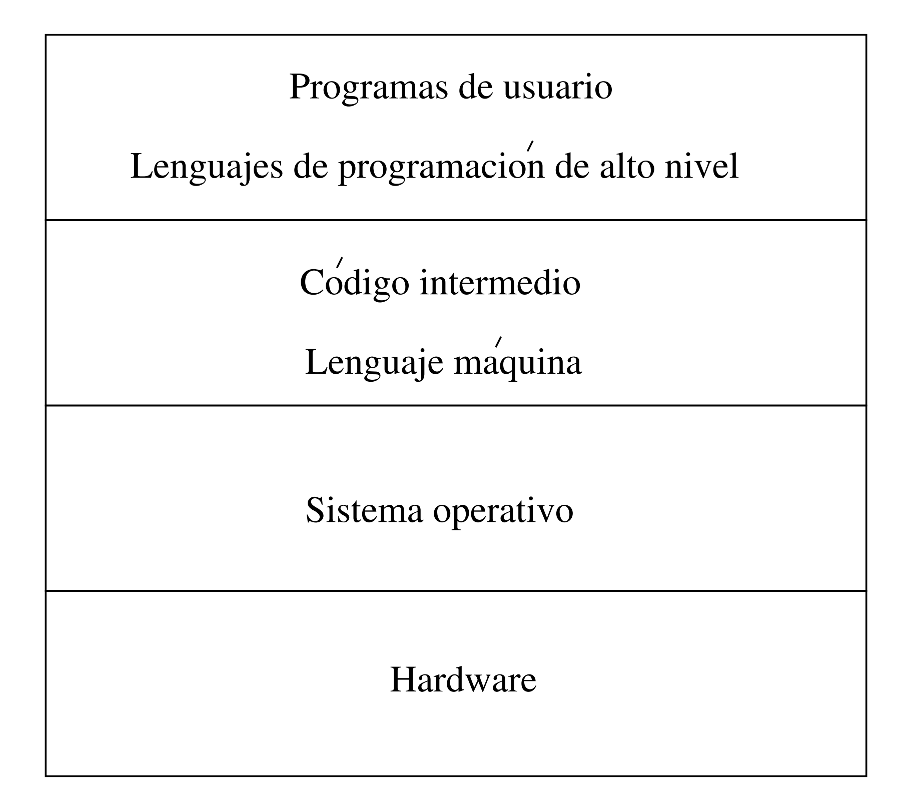


Figura 1.2. Esquema general que describe el funcionamiento del *software*: los programas de aplicación (o de usuario) se programan en lenguajes de alto nivel, que después son traducidos a lenguaje máquina. Es el sistema operativo quien se encarga de ejecutar finalmente los programas gestionando su acceso al hardware (memoria, procesador, disco duro. . .)

El sistema operativo (como veremos después) se encarga de ejecutar los programas y, para ello, los carga previamente en la memoria RAM. Ningún programa puede ejecutarse si no se carga en la memoria principal del ordenador. Por tanto, debe “caber” en ésta. Todo el proceso descrito se puede resumir en la Figura 1.2.

Veamos un ejemplo de programa escrito usando el lenguaje C que muestra un mensaje de bienvenida por la pantalla al ser ejecutado (previamente habrá que compilarlo, claro):

```
#include <stdio.h>

void main(char *argv[], int argc)
{
  printf("Saludos a todos.\n");
}
```

Existen centenares de lenguajes de programación disponibles en el mercado. En muchas ocasiones la elección de uno u otro para trabajar depende del tipo de problema que queremos resolver (aunque también depende de preferencias personales y de la "habilidad" personal de cada uno). Hay lenguajes más sencillos de entender y manejar que otros. Algunas indicaciones al respecto:

- Para programación de sistemas operativos o módulos (partes) del mismo o, incluso, controladores de dispositivos (*drivers*) los más utilizados son C y C++.
- Para labores de administración y mantenimiento de sistemas son muy adecuados lenguajes como Perl, Python o Awk.
- Para el desarrollo de aplicaciones en web, destacan PHP, Perl, Python y C#.
- Para programas de Inteligencia Artificial (robótica, juegos –ajedrez–, etc.) el lenguaje más utilizado es Lisp (también se utiliza Prolog).
- Si estamos pensando en lenguajes de propósito general, susceptibles de ser utilizados en casi cualquier situación, tenemos C, C++, Java, Pascal, Python y C#, por ejemplo.
- Otros lenguajes de programación, como Cobol, han caído en desuso y prácticamente sólo se utilizan en la programación de aplicaciones muy concretas en sistemas muy específicos.
- También está la "saga" *Visual* de Microsoft: Visual Basic, Visual C, Visual C++... muy orientados al desarrollo de aplicaciones de gestión en el entorno Windows. En este sentido, cabe destacar la nueva plataforma de programación desarrollada por Microsoft, .NET, y su lenguaje "estrella", C#. Precisamente, herramientas como *Visual Studio* y el API (*Application Program Interface*[6]) *Windows Forms* están orientadas al desarrollo "industrial" de programas con esta plataforma, la cual también se encuentra soportada en Linux con el proyecto *Mono* (aunque en menor medida).

Debes ser consciente que hay lenguajes más "potentes" que otros, más "portables" que otros, más "eficientes" que otros, etc. Para que te hagas una idea:

[6]Este término hace referencia, comúnmente, a un conjunto de librerías, funciones y/o programas de aplicación que pueden ser utilizados por los programadores en el desarrollo de nuevos programas y que contribuyen a aumentar, en buena medida, la productividad.

- Es muy difícil encontrar herramientas de trabajo (editores, compiladores) de lenguajes *Visual* fuera del entorno Windows. Además, estos lenguajes presentan una fuerte dependencia del sistema operativo (usan llamadas al sistema muy específicas de Windows). Ello quiere decir que los programas que desarrolles con ellas no podrán ser “migrados” (llevados a) otras arquitecturas. La excepción la constituye la plataforma .NET, la cual está soportada, por ejemplo, en Linux mediante *Mono*. Microsoft pretende potenciar el desarrollo de programas con esta plataforma y, por ello, se soporta en otros sistemas usados por los programadores. Lenguajes como C y C++ (sobre todo el primero) son mucho más portables ya que, además de ser estándares internacionales, es prácticamente imposible no encontrar herramientas de trabajo para C en cualquier arquitectura y/o sistema.

- Algunos lenguajes permiten expresar “más, con menos”, pero a costa de sacrificar cuestiones como la velocidad de ejecución de los programas o el consumo de memoria. Así, C, C++ o Pascal son lenguajes que permiten que los programas desarrollados con ellos se ejecuten rápidamente. Otros lenguajes como Python o Perl son más lentos, pero los programas que desarrollamos con ellos suelen ser entre 3 y 10 veces más cortos.

En la sección 1.3, al final de este capítulo, daremos una visión global de los diferentes tipos de lenguajes de programación, haciendo hincapié en sus características más destacadas, para que el estudiante pueda hacerse una idea somera de la capacidad y de la potencia del *software* como herramienta para solucionar problemas específicos (por ejemplo, en los campos de la inteligencia artificial, la informática gráfica, el reconocimiento de formas, la visión por computador, etc.).

1.2.3. Ficheros

Los datos que el ordenador maneja pueden residir en la memoria principal o en periféricos de almacenamiento masivo (dispositivos de almacenamiento secundario), como los discos duros, CD-ROM, etc. Por otro lado, los programas, demos, imágenes, etc., que se necesitan para el funcionamiento del ordenador, también tienen que estar guardados en discos duros, CD-ROM, etc. En el caso en que utilicemos periféricos de almacenamiento masivo para guardar datos y programas, la información se presenta organizada en *ficheros*. De esta manera podemos conservarlos para utilizarlos más tarde sin riesgo de que se pierdan al apagar el ordenador.

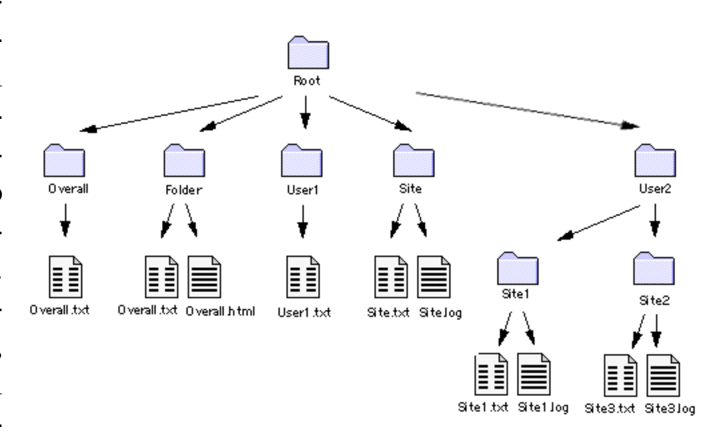


Figura 1.3. Organización jerárquica de un sistema de ficheros

Una definición formal de fichero, sería ésta: Un fichero es una colección de informaciones relacionadas entre sí y definidas por su creador. En general, un fichero es una secuencia de bits, bytes, líneas o registros cuyo significado ha sido definido por su creador y/o usuario. Los ficheros, por tanto, permiten:

- *agrupar* una serie de *datos* en una unidad lógica.
- *abstraer* de los *detalles de representación* y soporte físico del dispositivo sobre el que estamos trabajando. Trabajar con ficheros nos permite ignorar, por ejemplo, que, en realidad, en los discos duros la información se registra y lee mediante técnicas electromagnéticas y que en los CD-ROM se lee con medios ópticos (láser).

Podemos distinguir los siguientes tipos de fichero:

Ejecutables: sus datos son secuencias de instrucciones (programas) que el ordenador puede ejecutar.

De datos: guardan información de todo tipo (textos, imágenes, sonidos, bases de datos, hojas de cálculo. . .).

Un disco puede tener un elevado número de ficheros. Para organizarlos se agrupan en ciertas unidades, los *directorios* o carpetas. Dentro de un directorio puede haber, además de los ficheros, otros directorios. De este modo se crea una jerarquía, que se suele denominar *árbol de directorios*. Por ejemplo, en los sistemas de la familia UNIX existe una única raíz de la que cuelgan todos los directorios, en MS-DOS existe una raíz por cada disco y en la familia Windows más reciente (a partir del 95) cuelgan de *Mi PC*, pero conservando la identidad de los discos.

1.2.4. Sistemas operativos y entornos de usuario

Los programas de ordenador se pueden dividir en dos grupos principales:

- Programas del sistema (actividades propias del ordenador).
- Programas de aplicación (actividades de los usuarios).

El programa más importante de entre los del sistema es el *sistema operativo* (S.O.), el cual se encarga de controlar todos los recursos y proporcionar el soporte que necesitan los de aplicación. Es un programa que actúa de intermediario entre el usuario y el hardware del ordenador, permitiendo a aquél interactuar con éste ocultándole la complejidad de su manejo.

El usuario tiene la visión del ordenador a través del S.O., es decir, el usuario "ve" la imagen que el S.O. transmite del ordenador. Es por ello que resulta una parte fundamental del trabajo con ordenador, siendo muy importante su adecuada elección (en función de las tareas que deseamos realizar con él).

El S.O. es el programa básico que todo ordenador debe cargar primero en su memoria para poderse comunicar con el usuario. El S.O. se carga en el momento de encender la máquina y sirve de intérprete entre la máquina y el usuario humano. En síntesis, el S.O. es el gobierno interno del ordenador y se encarga fundamentalmente de:

- *Controlar y administrar los recursos* del ordenador. Distribuye los recursos del sistema (espacio en disco, memoria, tiempo de CPU) entre los usuarios y los programas, protegiendo a unos de otros.

  Objetivo: el funcionamiento eficiente del sistema informático, especialmente en el caso de sistemas multiusuario.

  ¿Para qué lo necesitamos? Pues, por ejemplo, para evitar que nuestros trabajos de impresión salgan mezclados con los de otros 25 usuarios.

- *Ofrecer una visión abstracta y homogénea*, independiente de la arquitectura, facilitando el manejo de los dispositivos y escondiendo la complejidad del hardware. Esto es, una *máquina virtual*.

  Objetivo: facilitar la utilización del ordenador por parte del usuario.

  ¿Para qué lo necesitamos? Por ejemplo para evitar tener que programar el acceso a un disco duro cada vez que queramos guardar información en él.

¿Sabías que...?

El S.O. también puede verse como un *programa de control*, en el sentido en que planifica la ejecución de los procesos, detectando y recuperando, en la medida de lo posible, los errores que pudiesen producirse.

Desde este punto de vista se pretende evitar el uso inadecuado de la máquina. El S.O. tiene la potestad de finalizar la ejecución de aquellos procesos que realizan operaciones "incorrectas". De esta forma, se evita que, por ejemplo, un programa produzca resultados anómalos debido a que otro ha modificado sus datos, accediendo inadecuadamente a la memoria que aquél ocupa.

El propósito del S.O. es proveer un ambiente en el cual el usuario pueda ejecutar programas y tener acceso a (manejar) los dispositivos de manera adecuada, cómoda y convenientemente. Como objetivo secundario está el de usar el hardware de una manera eficiente, permitiendo la compartición de recursos por los usuarios de forma (cuasi) óptima.

Desde nuestro punto de vista, hablaremos, más que de sistemas operativos, de entornos de trabajo o *entornos de usuario*, es decir, conjuntos de utilidades que, mediante una interfaz adecuada, permiten al usuario ejecutar programas, controlar procesos, gestionar sus ficheros y carpetas, etc. Un S.O. puede ofrecer al usuario más de un entorno de trabajo distinto y, además, de distinta naturaleza. *Grosso modo*, encontramos dos tipos de entornos de trabajo:

- de *línea de órdenes*, también conocidos como *intérpretes de órdenes*. En ellos el usuario escribe en un terminal o consola en modo texto las órdenes que desea que el sistema ejecute.

- *gráficos*, donde el usuario interactúa sobre elementos gráficos con uno o más dispositivos especiales (típicamente el ratón) para ejecutar las acciones deseadas.

Más cosas sobre los *procesos*

La unidad de trabajo (o carga) en un S.O. es el *proceso*. En general, un proceso es un programa en ejecución que se ejecuta secuencialmente (no más de una instrucción a la vez), aunque no necesariamente ha de existir una equivalencia entre ambos, ya que puede darse el caso de un programa único que, al ejecutarlo, lance varios procesos (pequeños programas auxiliares que ayudan a completar la ejecución del programa principal).

De hecho, el proceso es una entidad creada por el S.O. que se compone básicamente de:

- Código y datos del programa cargado en la memoria principal.
- Contexto en el que se ejecuta el programa, esto es, el estado del procesador en dicho momento.

Los procesos solicitan *recursos* al S.O. (memoria, acceso a dispositivos, ficheros. . .).

Estudiando las partes fundamentales que componen un S.O., podemos hacernos una idea de cuáles son sus funciones principales:

- Gestión de procesos ("programas"): Ejecución de programas y representación interna de los mismos en forma de procesos. Planificación: los procesos acceden a usar el procesador por turnos durante un tiempo limitado. Envío de señales a los procesos. Finalización de procesos a causa de errores.
- Entrada/salida: Resuelve la necesidades de sincronizar, atrapar interrupciones y ofrecer llamadas al sistema para los programadores, esto es, utilidades para que los programadores puedan capturar datos de entrada y mostrar resultados en sus programas.
- Gestión de la memoria: Asignación de memoria a los procesos que la solicitan. Liberar la memoria de los procesos que terminan. Llevar un control fidedigno del estado de la memoria principal en todo momento.
- Sistema de ficheros: Representación lógica del sistema físico de almacenamiento secundario (discos duros, disquetes, etc.). Gestión de los ficheros: correspondencia con sectores físicos. Precisamente, la operación de dar formato a un disco, partición o volumen construye el sistema de ficheros para el dispositivo, ya que permite establecer la correspondencia entre *unidades lógicas* y *unidades físicas*.

1.2.5. Redes

En general, se considera que una red de computadoras es cualquier conjunto de ordenadores que pueden comunicarse entre sí. Por lo tanto, una red puede abarcar únicamente dos ordenadores o varios millones. En muchos casos, distintas redes pueden estar comunicadas entre sí. Esto forma las "interredes" (*internets*), la más famosa de las cuales es la red de redes mundial conocida simplemente como *Internet*.

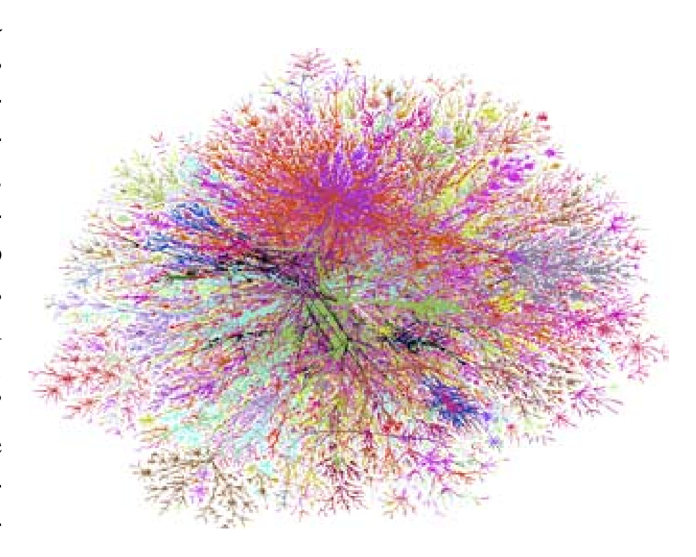


Figura 1.4. Esta imagen muestra los nodos principales de Internet y cómo estaban interconectados en el año 1998. El número de nodos, en la actualidad, es tan grande que sería imposible representarlos en una imagen del mismo tamaño

Entre las ventajas que proporcionan las redes de ordenadores, podemos destacar las que comentamos a continuación.

Si se trata de empresas:

- Permiten compartir recursos. Por ejemplo: el acceso a documentación, programas o máquinas situadas en lugares apartados geográficamente.
- Aumentan la fiabilidad. Por ejemplo: si los ficheros se mantienen en varias máquinas, cuando una de ellas falla se pueden emplear otras.
- Ahorran. La relación precio/rendimiento es mucho mejor en los ordenadores pequeños que en los grandes. Así, es preferible que cada usuario tenga su propio ordenador a que tenga una terminal de un ordenador grande. Esto ha extendido además el modelo cliente/servidor.
- Proporcionan escalabilidad, para obtener más potencia basta, en principio, con añadir nuevos ordenadores a la red.
- Además, son un medio de comunicación que permite trabajar juntos a empleados separados geográficamente.

Si se trata de personas:

- Permiten el acceso a información remota. Especialmente con el crecimiento de la *World Wide Web*.
- Permiten diversas maneras de comunicación interpersonal con servicios como el correo electrónico o la videoconferencia.

- Permiten nuevas maneras de entretenimiento (juegos en red).

Como hemos comentado, Internet es, en realidad, una red formada interconectando infinidad de redes de computadores.

1.2.6. Seguridad y protección de la información

Mucha de la información que almacenamos y procesamos en un ordenador es vital para sus propietarios y/o confidencial.

- ¿Cómo nos aseguramos razonablemente de no perderla cuando el ordenador falla? Necesitamos utilizar mecanismos *adecuados* para realizar *copias de seguridad* (*backups*).
- ¿Cómo nos aseguramos de que nadie accede a ella? Necesitamos utilizar mecanismos seguros para identificación de usuarios y ofrecer garantías de que las contraseñas no pueden ser descubiertas fácilmente.
- ¿Cómo nos aseguramos de que nadie la intercepta y lee durante una transmisión de un ordenador a otro? Necesitamos utilizar herramientas que garanticen la privacidad de la información que enviamos por la red: *criptografía*.
- ¿Cómo nos aseguramos de que nadie no autorizado accede a un ordenador y toma el control? Necesitamos utilizar mecanismos robustos que nos permitan ofrecer servicios al usuario sin tener que "abrir demasiadas puertas" para ello.

1.2.7. Aplicaciones

Una aplicación es un programa que facilita al usuario herramientas para llevar a cabo una tarea. Podemos clasificar las aplicaciones según el tipo de tarea que realizan:

Procesadores de texto: permiten preparar documentos para su posterior impresión. Debemos distinguir dos funciones principales:

Edición , que permite introducir, guardar en disco, recuperar e imprimir el contenido del texto, es decir, las palabras, frases, etc., que lo forman.

Formato , que permite fijar el aspecto (la forma) de aquello que se introduce.

Muchos procesadores de texto permiten ver el formato al mismo tiempo que se trabaja con el texto (*WYSIWYG*, What You See Is What You Get). Una "tentación" muy frecuente al usar procesadores de texto WYSIWYG (y casi, diríamos, inherente a su uso) consiste en "olvidarse" del contenido para centrarse en la forma. Se recomienda postergar hasta el último momento, en la medida de lo posible, la mayor parte de decisiones relativas al formato del documento. Llegado este punto, permítasenos dar un consejo al lector: el documento no debe ser un "festival" de demostración de las posibilidades del procesador, no hay que abusar de tipos de letra y demás efectos.

Editores de texto: fundamentalmente para edición de “contenidos” sin opciones de formato. Se utilizan habitualmente para escribir y editar programas, control total de la edición de páginas web...

Suelen proporcionar herramientas que ayudan a escribir programas o bien el tipo de documentos con el que se trabaja y poseen funciones de edición avanzadas, como, por ejemplo, coloreado de las palabras reservadas de un lenguaje de programación, sangrado automático, etc.

Dibujo vectorial: sirven para dibujar objetos almacenando la información necesaria para construirlos (por ejemplo, de una circunferencia se puede almacenar su radio, las coordenadas del centro y el grosor y el trazo de la línea que la dibuja).

Permiten mayor flexibilidad en el dibujo; se manipulan atributos de los objetos permitiendo escalados, giros, estrujamientos, modificaciones en el relleno, etc. Están orientados al dibujo de precisión: diseño de planos, diseño gráfico... Por sus características es fácil modificar objetos y crear otros más complejos por composición de objetos simples. Los objetos suelen tener gran variedad de atributos susceptibles de ser modificados: contorno, relleno, tipo de trazo, etc.

Retoque fotográfico: permiten modificar y editar una imagen fotográfica o partes de la misma. Poseen varias funcionalidades para resaltar, ampliar, etc., una imagen. Se utilizan para fotografía digital, diseño, publicidad, etc.

Programas de CAD/CAM: son programas de dibujo vectorial con capacidades muy avanzadas: i) permiten la manipulación de objetos en tres dimensiones; ii) se pueden utilizar para generar distintos tipos de proyecciones (planos) y perspectivas de un objeto; iii) se pueden integrar con programas de fabricación automática (CAM). Se utilizan mucho en ingeniería y en diseño y fabricación.

Bases de datos: los sistemas de bases de datos permiten guardar grandes cantidades de información relacionadas entre sí. Para ello se definen las características de esos datos, las formas de acceder a ellos y las relaciones “estructurales” que tienen entre sí. Por ejemplo: un registro podría tener los datos de un empleado, entonces los campos serían informaciones como su nombre, DNI (clave), fecha de nacimiento, dirección, teléfono, etc. Las relaciones entre unos registros y otros se realizan a partir de las claves.

Programas para presentaciones: estos programas permiten crear una serie de “pantallas” (“animadas” o no) para hacer presentaciones. Una vez creadas las pantallas tenemos varias opciones: i) presentarlas, una a una, en pantalla; ii) imprimirlas, normalmente para crear transparencias; iii) imprimirlas para hacer diapositivas. Además, se dan facilidades para crear documentación adicional y para distribuir o para guardar notas de la presentación.

Hojas de cálculo: son disposiciones matriciales en forma de filas y columnas de *celdas*. En cada celda podemos introducir información (literal, numérica o especiales como fechas y horas) o fórmulas para relacionar el valor de la celda con los

de otras. De esta manera se simula un "estadillo" que se actualiza instantáneamente. Esto permite simplificar cálculos repetitivos y hacer comprobaciones del tipo "what-if" (¿qué pasaría si…?). Además permiten representar gráficamente los valores almacenados en un rango (un área) de celdas.

Navegadores: técnicamente hablando son *intérpretes* de código HTML, es decir, permiten visualizar ficheros escritos usando este lenguaje de etiquetas que es el que se utiliza en la elaboración de páginas web. También permiten acceder a ficheros de vídeo, audio, imágenes, PDF, etc., usando los *plugins*[7] adecuados. Algunos, como Internet Explorer y Konqueror, se "integran" en entornos gráficos de trabajo y permiten acceder al contenido de ficheros y carpetas del sistema.

Correo electrónico: son herramientas que permiten descargar, leer, editar y enviar correo electrónico (si tenemos conexión a Internet).

Otros programas: herramientas de ayuda a la programación: compiladores, intérpretes, depuradores de código, etc.; programas de creación y manipulación de sonido; juegos; programas de control de maquinaria; etc.

1.3. LENGUAJES DE PROGRAMACIÓN

La programación de las primeras máquinas computadoras resultaba enormemente complicada, ya que había que conmutar circuitos mediante el empleo de conexiones cableadas y llaves mecánicas, lo cual hacía en la práctica casi imposible la tarea de realizar programas, entendiendo como tal el almacenamiento por medios electrónicos de un conjunto de instrucciones que debe ejecutar una computadora para llevar a cabo una tarea.

Posteriormente, con el advenimiento de los primeros computadores de propósito general, ENIAC (*Electronic Numerical Integrator And Computer*; Computador e Integrador Numérico Electrónico, en castellano) y, su sucesor, UNIVAC[8] (*UNIVersal Automatic Computer*; Computadora Automática Universal, en castellano), se sentaron las bases que posteriormente darían lugar a los primeros lenguajes de programación simbólicos.

La primera generación de computadores UNIVAC fue diseñada para su uso en tareas administrativas, es decir, para la ejecución rápida de grandes cantidades de operaciones aritméticas relativamente simples y transporte de datos, a diferencia de los cálculos numéricos complejos requeridos por las computadoras científicas, tales como ENIAC.

De hecho, UNIVAC competía directamente con las máquinas de tarjeta perforada hechas principalmente por IBM, aunque inicialmente no dispuso de interfaz para la lectura o perforación de tarjetas. Luego, se dotó a UNIVAC de sendos convertidores, de tarjeta a cinta y de cinta a tarjeta, para la transferencia de datos entre las tarjetas y las cintas magnéticas que empleaba UNIVAC de forma nativa.

[7] Herramientas específicas para el tratamiento adecuado de un determinado tipo de ficheros.

[8] De este modelo de computador se fabricaron, en realidad, dos generaciones distintas (I y II). Aquí nos estamos refiriendo a la primera generación.

Estos primeros computadores se programaban directamente en código binario, es decir, mediante secuencias de unos y ceros (sistema binario para la codificación de la información, que estudiaremos con detalle en el capítulo 3). Cada modelo de ordenador desarrolló su propio código, es decir, su propio *juego o conjunto de instrucciones*, donde a cada operación básica le correspondía una determinada secuencia de ceros y unos (los datos con los que se operaba también se representaban en binario). Es por ello por lo que a estos "lenguajes" se les denominó *lenguajes máquina* (o también código máquina). Los lenguajes máquina se han considerado históricamente como la primera generación de lenguajes de programación.

A la hora de programar los computadores, la secuencia de cableado de conexiones representaba, en realidad, una secuencia de unos y ceros que venía a representar una instrucción o un conjunto de instrucciones (programa) para el computador en el que se estaba trabajando. Posteriormente, las tarjetas y las cintas perforadas, que utilizaban la misma lógica (perforación o ausencia de perforación para representar unos y ceros), se utilizaron como medio de almacenamiento de programas y también como mecanismo para la introducción de datos.

Hace ya mucho tiempo que las tarjetas perforadas cayeron en desuso, siendo sustituidas por otros medios (magnéticos y ópticos) para almacenar datos. Como curiosidad, quisiéramos apuntar que algunos medios actuales de almacenamiento de datos, como es el caso de los CD-ROM, se basan en un método similar al usado por las tarjetas perforadas (realizando diminutas muescas sobre una superficie de material liso, lo cual hace que se refleje o no la luz de un láser que incide sobre ella), aunque, por supuesto, los tamaños, velocidades de acceso y capacidad de los medios actuales no admiten comparación con las viejas tarjetas.

Programar en lenguaje máquina continuaba siendo una tarea tediosa. Era muy difícil elaborar programas considerados hoy en día como sencillos y, más difícil aún, depurarlos. La solución que se pensó alumbró lo que se conoce como la segunda generación de lenguajes de programación, constituida por los denominados *lenguajes ensambladores*.

Cuando se programaba en lenguaje máquina, uno de los mayores inconvenientes residía en la dificultad de recordar las secuencias de unos y ceros que representaban una determinada operación del juego básico de instrucciones. Era obligado consultar casi continuamente la documentación para elaborar un programa. La idea que surgió para solucionar este problema era muy sencilla: se decidió denominar a las operaciones usuales en la programación de los computadores con nombres fáciles de memorizar y asociar. Por ejemplo, ADD para la suma, SUB para la resta, MUL para la multiplicación, o CALL para la llamada a (ejecución de) una subrutina. Todos estos nombres o *etiquetas* constituyeron las instrucciones de lo que se conoció como *lenguaje ensamblador*.

Aunque la programación de los computadores existentes en la época era notoriamente más sencilla y productiva gracias al empleo de los lenguajes ensambladores –ya que simplificaban la escritura de las instrucciones y hacían los programas más legibles–, todavía aparecían serios inconvenientes para la programación de propósito general. Ciertamente, el uso de los lenguajes ensambladores hacía más fácil la depuración de los programas y, realmente, su traducción a lenguaje máquina (el único que entendían y entienden los computadores, o mejor dicho, sus procesadores) era casi directa (bastaba

con sustituir las etiquetas por el código binario de las instrucciones correspondientes consultando una tabla). Pero los lenguajes ensambladores eran propios y, por tanto, dependientes del tipo de computador para el cual se habían desarrollado. Esto imposibilitaba la *portabilidad* de los programas: era necesario reescribir el programa si se quería ejecutarlo en otro tipo de computador, con lo cual la productividad de los programadores se resentía en gran medida.

En general, podemos afirmar que la evolución de los lenguajes de programación corrió en paralelo con la de los computadores, y el *hardware* en general, durante buena parte del siglo pasado. De hecho, los lenguajes de programación suelen clasificarse en cinco generaciones (ya hemos visto las dos primeras), de las que las tres primeras son evidentes, mientras no todo el mundo está de acuerdo en las otras dos. Estas sucesivas evoluciones de los lenguajes de programación no coincidieron exactamente en el tiempo con los grandes hitos en el desarrollo del *hardware*, pero sí de forma aproximada.

Posteriormente, con la llegada de los años sesenta y el desarrollo de la llamada segunda generación de computadores, caracterizados por usar circuitos con transistores en vez de válvulas al vacío, aparecieron diferentes lenguajes de programación, los cuales reciben su denominación porque tienen una estructura sintáctica similar a los lenguajes escritos por los humanos.

Así comienza el desarrollo de la denominada tercera generación en la evolución de los lenguajes de programación. Los *lenguajes de alto nivel* sustituyen las instrucciones simbólicas dependientes de la máquina de los lenguajes ensambladores por códigos independientes del computador, empleando instrucciones parecidas al lenguaje humano o, mejor dicho, al de las matemáticas. De esta manera, surgió la primera hornada de lenguajes de alto nivel considerados de propósito general, aunque ya veremos que esto no es del todo cierto:

- FORTRAN (1957): este lenguaje de estilo procedural fue el primero de alto nivel, desarrollado por IBM en 1957 para el equipo IBM 704. Está fuertemente orientado al cálculo, siendo uno de los de mayor eficiencia en la ejecución. Aunque la definición estándar del lenguaje no se establecería hasta mediados de los sesenta, sufrió varias revisiones (versiones) siendo empleado con cierta notoriedad, fundamentalmente en cálculo científico, hasta finales de los noventa.

  Cabe destacar que las líneas de los programas deben estar numeradas (por su orientación inicial para el almacenamiento en tarjetas perforadas de 80 columnas) y que, inicialmente, la única alteración posible en el flujo de los programas venía dada por el empleo de la instrucción `goto`. Luego, en versiones posteriores, aparecieron otras estructuras de control tales como subprogramas y recursión. Hoy en día se considera un lenguaje obsoleto que ha desaparecido del TOP 20 en todos los rankings con las estadísticas de uso de lenguajes de programación (`http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html`).

- LISP (1958): es un lenguaje de programación de tipo declarativo que fue creado en 1958 por John McCarthy y sus colaboradores en el MIT. Es un lenguaje orientado al paradigma de la *programación funcional*.

El objetivo de la programación funcional es conseguir lenguajes expresivos desde el punto de vista matemático, en los que no sea necesario bajar al nivel del *hardware* (por ejemplo, manejando directamente posiciones de memoria RAM, como es el caso de C) para describir el proceso llevado a cabo por el programa, y evitando el concepto de estado de ejecución[9]. La secuencia de cálculos llevada a cabo por el programa se regiría única y exclusivamente por la reescritura de definiciones más amplias a otras cada vez más concretas y definidas.

Esto quiere decir que, básicamente, los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones, entendiendo éstas no como subprogramas clásicos de un lenguaje imperativo, sino como funciones puramente matemáticas, en las que se verifican ciertas propiedades. Otras características propias de estos lenguajes son la no existencia de asignaciones de variables y la falta de construcciones estructuradas como la secuencia o la iteración (lo que obliga en la práctica a que todas las repeticiones de instrucciones se lleven a cabo por medio de funciones recursivas).

El elemento fundamental en LISP es la lista, en el sentido más amplio del término, pues tanto los datos como los programas son listas, las cuales vienen delimitadas por paréntesis. De ahí viene su nombre, pues LISP es un acrónimo del inglés *LISt Processing*. LISP sigue una filosofía de tratamiento no destructivo de los parámetros del programa, de modo que la mayoría de las funciones devuelven una lista resultado de efectuar alguna transformación sobre la que recibieron, pero sin alterar esta última.

Uno de los motivos por los que LISP resulta especialmente adecuado, y continúa siendo un lenguaje utilizado, es el hecho de que el código y los datos tengan el mismo tratamiento (como listas); esto hace especialmente sencillo escribir programas capaces de escribir otros programas según las circunstancias. Concretamente, en el campo de la *Inteligencia Artificial* LISP sigue siendo un lenguaje popular.

- ALGOL (1960): es un acrónimo de las palabras inglesas *ALGOrithmic Language* (lenguaje algorítmico, en castellano). Fue muy popular en las universidades durante los años sesenta, pero no llegó a cuajar como lenguaje de utilización comercial. Su importancia reside en el hecho de que se le considera el precursor de varios lenguajes posteriores que sí que alcanzaron gran difusión, tales como Pascal o C.

- COBOL (1960): es un acrónimo de *COmmon Business-Oriented Language* (en castellano, lenguaje común orientado a los negocios). Fue creado en el año 1960 con el objetivo de desarrollar un lenguaje de programación universal que pudiera ser usado en cualquier ordenador, ya que en los años sesenta existían numerosos modelos de ordenadores incompatibles entre sí, y que estuviera orientado principalmente a los negocios, es decir, a la llamada informática de gestión.

  Entre los puntos fuertes de COBOL está la gestión de ficheros y de los tipos de

[9]Que puede definirse, a grandes rasgos, como el valor que tienen todas las variables empleadas para almacenar datos en el programa en un instante determinado.

datos (registros). Fue un lenguaje muy utilizado en las décadas de los sesenta, setenta y ochenta. Pese a esto, a comienzos de los ochenta, se fue quedando anticuado respecto a los nuevos paradigmas de programación y a los lenguajes que los implementaban. En la revisión que sufrió el lenguaje en 1985 estas carencias se solucionaron en parte, incorporando a COBOL variables locales, recursividad, reserva de memoria dinámica y programación estructurada.

A principios de este siglo, se incorporaron nuevas funcionalidades al lenguaje que han evitado su obsolescencia. Aunque parezca mentira, sigue siendo un lenguaje más utilizado de lo que se piensa, ya que muchas entidades bancarias y otras grandes empresas todavía disponen de sistemas *mainframes* (véase el capítulo 4) que disponen de compiladores y herramientas de desarrollo para COBOL.

La mayoría de estos lenguajes son *compilados*. Una vez escrito el programa se requiere el empleo de un programa específico, denominado compilador, que traduzca el programa a lenguaje máquina listo para su ejecución en el procesador del ordenador con el que se esté trabajando. La portabilidad de los programas escritos en un determinado lenguaje de programación de alto nivel a otros tipos de computador queda asegurada si se dispone del correspondiente programa compilador del lenguaje.

Con posterioridad al nacimiento de estos primeros lenguajes y, sin duda influidos por su desarrollo, aparecieron nuevos lenguajes de programación a lo largo de las décadas de los años sesenta y setenta. Entre ellos, consideramos que merece la pena destacar por su relevancia, trascendencia e influencia, los siguientes: BASIC, RPG[10], Pascal, C y PROLOG[11].

BASIC, acrónimo de *Beginners All-purpose Symbolic Instruction Code* (código de instrucciones simbólicas de propósito general para principiantes, traducido al español) es un lenguaje de programación que, inicialmente, fue pensado para su empleo como herramienta de enseñanza con el fin de aprender a desarrollar fácilmente programas básicos y simples. Posteriormente, y por razones comerciales, se diseminó enormemente entre los adolescentes y los jóvenes, en general, interesados en el mundo de la informática. Originalmente, BASIC fue un lenguaje compilado, aunque posteriormente, con sucesivas versiones del lenguaje, derivó en un *lenguaje interpretado*, es decir, que requería que un programa específico, llamado intérprete, fuese el que ejecutase los programas hechos en este lenguaje instrucción por instrucción (el intérprete traduce a lenguaje máquina una instrucción, ésta es ejecutada por el procesador del ordenador y, a continuación, procede de la misma forma con la siguiente).

Las razones comerciales a las que aludíamos en el párrafo anterior tienen que ver con el hecho de que muchos de los primeros microordenadores caseros comercializados entre finales de los setenta y principios de los ochenta (como los populares Altair 8800 y ZX-Spectrum) incluían un intérprete de BASIC. Esto ocurrió ya que la mayoría de los lenguajes de programación eran demasiado complejos para que sus

[10]Sí, entendemos que puede parecer extraño destacar este lenguaje un tanto arcaico, pero curiosamente (y ya veremos por qué) es un lenguaje de programación que continúa usándose en la actualidad en buena medida.

[11]Como estandarte del paradigma de la programación declarativa, de importancia en algunas áreas de la Inteligencia Artificial, tales como el desarrollo de sistemas expertos.

compiladores pudieran ser albergados por las pequeñas memorias que la mayoría de microordenadores de la época tenían. Si, además, tenemos en cuenta el lento almacenamiento que proporcionaba la cinta de papel y, más tarde, las cintas de audiocasete (los discos magnéticos aún no existían), y la falta de editores de texto adecuados, un lenguaje sencillo como BASIC era una buena opción. También, el primer sistema operativo desarrollado para PC (*Personal Computer*; ordenador personal, en castellano), MS-DOS (*MicroSoft Disk Operating System*), traía incorporado un intérprete de BASIC desarrollado por la misma empresa (Microsoft).

BASIC fue diseñado para permitir a los estudiantes escribir programas usando terminales de computador de tiempo compartido. Estaba pensado para reducir notablemente la complejidad de los otros lenguajes del momento, con uno nuevo, diseñado específicamente para la clase de usuarios que los sistemas de tiempo compartido permitían: un usuario más sencillo, a quien no le interesaba tanto la velocidad, sino el hecho de ser capaz de programar y usar la máquina sin demasiadas complicaciones. Los diseñadores del lenguaje también querían que permaneciera en el dominio público, lo cual contribuyó a que se diseminara rápidamente.

Posteriormente, en los años ochenta y noventa aparecieron varias extensiones de BASIC para ordenadores personales con capacidad para desarrollar gráficos y sonido, así como facilidades para programación estructurada. Hoy en día, BASIC continúa siendo un lenguaje muy popular, aunque eso sí, pervive a través de varios dialectos de reciente desarrollo bastante distintos del original, tales como Visual Basic y .NET Visual Basic, ambos de Microsoft.

El lenguaje de programación RPG fue desarrollado por IBM en 1964. Originalmente, su diseño estaba claramente enfocado a la generación de informes comerciales y de negocios (por ejemplo, normalmente, informes que procesaban registros de una base de datos y mostraban los resultados pedidos). Sus siglas en inglés significan *Report Program Generator* (en castellano, Programa Generador de Informes).

RPG es un lenguaje con un flujo de cómputo prefijado, que no sigue las directrices de la programación estructurada (posteriormente comentaremos algo al respecto). Es un lenguaje un tanto arcaico que, sin embargo, sigue siendo bastante utilizado porque muchas empresas aún disponen de sistemas IBM AS/400, que utilizan este lenguaje como herramienta para el proceso de datos.

Este tipo de *mainframes* (véase el capítulo 4) fueron desarrollados por IBM en los años noventa, como sucesores de la generación IBM 360. Los IBM 360 también usaban RPG para el proceso de datos e IBM decidió mantener esta característica en su nueva gama por compatibilidad. Posteriormente, en 2001, y con la aparición de la versión 5 del AS/400, surgió una nueva modificación del lenguaje, soportándose a partir de ese momento la programación en formato libre (eliminando el flujo de cómputo prefijado). También, se desarrollaron funciones incorporadas que sustituyen a muchos de los antiguos indicadores y códigos de operación. Todos estos nuevos diseños han permitido que RPG se convirtiese en un lenguaje mucho más legible, claro, flexible y moderno.

En un artículo de 1968[12], Edsger Dijkstra, el considerado fundador de los princi-

[12] DIJKSTRA, E. W. (1968): “Go To Statement Considered Harmful”, Communications of the ACM, vol. 11, pp. 147–148,

pios de la programación estructurada, afirmaba que los lenguajes de programación que usaban sentencias GOTO para estructurar el flujo de ejecución de un programa eran nocivos para la productividad del programador, y para la calidad del código resultante. En este artículo no se menciona ningún lenguaje de programación en particular (aunque muchos consideraron que BASIC era el objetivo fundamental de sus críticas, ya que muchos de sus programadores tenían una marcada tendencia a usar el GOTO para el desarrollo de bucles); únicamente se indicaba que el uso excesivo de GOTO era algo no recomendable, y proporcionaba las razones técnicas por las que esto es así.

Al mismo tiempo, Dijkstra propugnó algunas prácticas para el desarrollo de programas "de calidad" (legibles, fáciles de depurar y bien estructurados), que fueron conocidas como los principios de la denominada *programación estructurada*. Entre ellos, por su influencia en el desarrollo de lenguajes posteriores, cabe destacar:

- La declaración de variables y constantes al inicio del programa.
- La determinación de los tipos de variable (número, texto. . .).
- La utilización de instrucciones que permitan la ejecución de bucles ligados al cumplimiento de ciertas condiciones (sin emplear saltos incondicionales, tipo GOTO).
- La definición de subrutinas para agrupar instrucciones que realizan una misma tarea.

Fruto de estos principios de la programación estructurada, aparecieron, a principios de los setenta, lenguajes relevantes tales como Pascal y C. Pascal es un lenguaje de programación que fue desarrollado por el profesor suizo Niklaus Wirth en 1970. Su objetivo era crear un lenguaje que facilitara el aprendizaje de la programación a sus alumnos. Sin embargo, con el tiempo, su utilización excedió el ámbito académico para convertirse en un lenguaje de propósito general que permite la creación de programas de todo tipo.

Pascal se caracteriza por ser un lenguaje de programación estructurado fuertemente tipificado. Esto implica que, por un lado, el código está dividido en porciones fácilmente legibles, llamadas funciones o procedimientos. De esta forma, Pascal facilita la utilización de la programación estructurada en oposición al antiguo estilo de programación monolítica. Por otro lado, el tipo de dato de todas las variables debe ser declarado previamente para que su uso quede habilitado en el programa. El nombre de Pascal fue escogido en honor al matemático Blaise Pascal.

En los años ochenta y noventa, Pascal fue un lenguaje muy utilizado, sobre todo gracias al desarrollo de sucesivas versiones, entre las que cabe destacar la serie TurboPascal. TurboPascal fue un lenguaje de programación que, en sus primeras revisiones, se apartaba en cierta medida del Pascal estándar, pero que luego derivó en un lenguaje estándar que incluía librerías gráficas y otras utilidades orientadas al desarrollo de programas comerciales.

Más recientemente, y como derivado del *Object Pascal*, que, a su vez, deriva de una ramificación de TurboPascal 5.5, apareció el entorno de desarrollo *Delphi*, que puede considerarse como una versión de cuarta generación del lenguaje original, aunque realmente ya sólo quedan reminiscencias.

C es un lenguaje de programación que fue creado en 1972 por Kenneth L. Thompson y Dennis M. Ritchie, que trabajaban en los Laboratorios Bell y que crearon, en paralelo, el sistema operativo Unix (véase el capítulo 2). C fue inicialmente un lenguaje orientado a la implementación de sistemas operativos, concretamente del sistema operativo Unix. Posteriormente, y debido a su potencia, flexibilidad y portabilidad, se consideró como un lenguaje de propósito general más y, por tanto, fue empleado para el desarrollo de programas de todo tipo, incluso *software comercial*.

La enorme portabilidad de C se debió, en sus orígenes, al gran éxito comercial del sistema operativo Unix y de los computadores que lo utilizaban. Inicialmente, el código (las instrucciones que conforman el programa) del sistema operativo Unix estaba disponible para su estudio por parte de las universidades y centros de investigación, es decir, era un sistema operativo *open source* (de "fuente abierta", es decir, que si queremos podemos acceder al código). Al extenderse rápidamente su uso por empresas, centros de investigación y universidades, también se extendió el uso del lenguaje C, ya que al estar Unix programado en C, era obligado utilizar este lenguaje para programar el sistema y desarrollar utilidades y programas de usuario.

Posteriormente, se desarrollaron otros sistemas operativos comerciales derivados de Unix, tales como AIX (IBM) y HP-UX (Hewlett Packard), que, aunque no eran *open source*, contribuyeron en gran medida a difundir enormemente el uso de C. En los años ochenta y noventa, prácticamente se podía afirmar que no había ninguna arquitectura (tipo de ordenador) que no incluyese un compilador de C. A ello contribuyó, sin duda, la estandarización del lenguaje (ANSI; del inglés *American National Standards Institute*, Instituto Nacional Estadounidense de Estándares, en castellano), con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia, por lo que, si los programas creados lo siguen, el código es portátil entre plataformas y/o arquitecturas. En la práctica, los programadores suelen usar elementos no portables dependientes del compilador o del sistema operativo, aunque eso afecta, normalmente, a un porcentaje pequeño del código.

C es un lenguaje de alto nivel que incorpora las estructuras típicas de los lenguajes afines a los principios de la programación estructurada, pero que, al contrario que Pascal, es débilmente tipificado, ya que aunque obliga a definir los tipos (número entero, número real, texto. . .) de las variables empleadas en el programa, no es necesario definir éstas al principio del mismo. Además, el lenguaje permite, en principio, la asignación de valores entre variables de tipos distintos, realizando, de forma transparente, las transformaciones pertinentes, lo cual suele ser fuente de errores de programación difíciles de depurar.

C también tiene muchas características de los lenguajes de bajo nivel, como el lenguaje ensamblador, ya que dispone de construcciones del lenguaje que permiten un control a muy bajo nivel: por ejemplo, los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclarlo con código en ensamblador, y el lenguaje incluye funcionalidades que permiten acceder directamente a memoria o a dispositivos periféricos.

C es un lenguaje apreciado por los profesionales, debido a la eficiencia del código que produce, y, actualmente, se considera el lenguaje de programación más popular

para crear *software* de sistemas (programas manejadores de dispositivos, lo que se conoce por el anglicismo *drivers*), aunque también se utiliza para crear otras aplicaciones (pero cada vez menos).

PROLOG es un lenguaje de programación lógica (del francés *PROgramation en LOGique*) ideado a principios de los años setenta en la universidad de Aix-Marseille por los profesores Alain Colmerauer y Phillipe Roussel. Inicialmente se trataba de un lenguaje totalmente interpretado hasta que, a mediados de los setenta, David H. D. Warren desarrolló un compilador capaz de traducir Prolog en un conjunto de instrucciones de una máquina abstracta denominada Warren Abstract Machine, o abreviadamente, WAM.

Los lenguajes de programación lógica están basados en el razonamiento formal. Los objetos de tales razonamientos son "hechos" o reglas "if then" (si-entonces) que utilizan un conjunto de hechos para calcular la verdad o falsedad de ese conjunto de hechos. Un hecho es una expresión en la que algún objeto o conjunto de objetos satisface una relación específica.

Los programas en Prolog se componen de *cláusulas de Horn*[13], que constituyen reglas del tipo "modus ponendo ponens", es decir, "si es verdad el antecedente, entonces, es verdad el consecuente". No obstante, la forma de escribir las cláusulas de Horn es al contrario de lo habitual, esto es, primero se escribe el consecuente y luego el antecedente. El antecedente puede ser una conjunción de condiciones que se denomina secuencia de objetivos. Cada objetivo se separa con una coma y puede considerarse similar a una instrucción o llamada a procedimiento de los lenguajes imperativos.

En Prolog no existen instrucciones de control (por ejemplo, instrucciones condicionales asociadas al cumplimiento de ciertos hechos o bucles). Su ejecución se basa en dos conceptos: la unificación y la búsqueda predeterminada de soluciones en el espacio de todos los hechos coherentes con el conjunto de reglas especificado. Prolog es un lenguaje que sigue empleándose en la actualidad, fundamentalmente en algunas áreas de la Inteligencia Artificial (por ejemplo, en el área de sistemas expertos basados en reglas).

Los lenguajes estructurados desarrollados en los setenta, fundamentalmente Pascal y C, marcaron el paso de la programación durante los años ochenta y noventa; aunque seguían sin colmar las ansias de productividad de los programadores profesionales. Precisamente, y con el fin de aumentar la productividad de los programadores y la reusabilidad del código, surgió el paradigma de la *programación orientada a objetos*, cuyos representantes más destacados fueron, inicialmente, los lenguajes C++ y Java.

La programación orientada a objetos (POO) asume los principios de la programación estructurada, pero va un paso más allá. En lugar de centrarse en la definición de subrutinas, es decir, funciones y procedimientos que agrupan código (secuencias de instrucciones) cuyo propósito es realizar una acción determinada, la POO se basa fundamentalmente en construir las estructuras de datos (objetos) que requiere nuestro programa y en definir las acciones y métodos a los cuales dichos objetos deben responder. Por tanto, bajo este paradigma, un programa se estructura en base a la definición de objetos y a su interacción según la lógica definida por la aplicación a desarrollar. Algunos de los principios fundamentales de la POO son:

[13] `http://es.wikipedia.org/wiki/Clausulas_de_Horn`.

- Encapsulamiento: la estructura interna que define a un objeto sólo puede accederse a través de los métodos que lo manipulan.
- Herencia: es la facilidad mediante la cual una clase hereda total o parcialmente los atributos (estructura interna) y métodos definidos en otra, para, a continuación, añadir sus atributos y métodos propios. Esto favorece enormemente la reutilización de código.
- Polimorfismo: es una característica de los lenguajes OO que permite que un mismo método pueda comportarse de forma distinta en función de la clase del objeto sobre el que actúa. También puede ocurrir que un método que actúa sobre los objetos de una clase pueda ser invocado de diversas maneras (por ejemplo, con un número distinto de parámetros). Esto último también se conoce como sobrecarga de operadores.

C++ fue el primer lenguaje *popular* (con un uso extendido, queremos decir) que usaba los principios de la POO. C++ es un lenguaje de programación que fue diseñado a mediados de los años ochenta por Bjarne Stroustrup. La intención de su creador fue incorporar mecanismos que permitiesen la manipulación de objetos al lenguaje de programación C. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, a C++ se le considera un lenguaje híbrido, ya que, en puridad, no implementa todos los mecanismos que requiere el paradigma de la POO. Se suele decir que C++ es un lenguaje multiparadigma ya que incorpora algunos de los mecanismos contemplados en la orientación a objetos junto con los principios de la programación estructurada que "trae de serie" al ser un derivado de C.

Actualmente existe un estándar, denominado ISO C++, al que se han adherido la mayoría de los fabricantes de compiladores más modernos. Una particularidad de C++ es la posibilidad de redefinir los operadores existentes en el lenguaje (sobrecarga de operadores), y de poder crear nuevos tipos que se comporten como tipos fundamentales. También cabe destacar el hecho de que, al ser un derivado de C, permite trabajar tanto a alto como a bajo nivel, ya que incluye funcionalidades para el acceso a memoria y dispositivos.

Pero, sin duda, el lenguaje de programación orientado a objetos más popular, y el más utilizado en la actualidad (por encima de C y C++), es Java, que fue desarrollado a principios de los noventa por la multinacional informática Sun Microsystems. Algunos lo consideran como un miembro más de la "familia del C", ya que el lenguaje en sí mismo toma mucha de su sintaxis de C y C++, aunque cabe decir que tiene un modelo de objetos más simple y elimina las herramientas de bajo nivel, que suelen inducir a muchos errores de programación, como son la manipulación directa de direcciones y posiciones de memoria. También cabe destacar que su modelo de ejecución, a través de una *máquina virtual*, es distinto del modelo de C y C++, basado en la generación de ejecutables en el lenguaje máquina destino.

La máquina virtual Java es una aplicación (que podemos ver como un programa o conjunto de programas) que proporciona una especie de entorno operativo donde se interpretan las instrucciones del programa escrito en Java que deseemos ejecutar. Por tanto, en principio, la ejecución de cualquier programa Java se realiza a través de la máquina virtual Java. Para ello, el programa se compila previamente a una especie de

código máquina de alto nivel, propio de la máquina virtual, que se conoce popularmente como *bytecode*. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo (código máquina del procesador) mediante la máquina virtual para su ejecución definitiva. No obstante, quisiéramos resaltar que es posible llevar a cabo tanto la compilación en código máquina nativo de un programa Java, como la ejecución directa por *hardware* del *bytecode* a través de un procesador Java.

La portabilidad de Java depende de su máquina virtual. En aquellas plataformas en las que haya disponible una versión de la máquina virtual Java se podrá ejecutar los programas Java desarrollados sin mayor problema. Ésta es una de las mayores ventajas de Java. La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del *Java Community Process*, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias tipo *open source*.

Entre noviembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GPL de GNU (véase el capítulo 7), de acuerdo con las especificaciones del *Java Community Process*, de tal forma que prácticamente todo el Java de Sun es ahora *software* libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java todavía no lo es).

No quisiéramos terminar nuestro breve recorrido por los lenguajes orientados a objetos empleados con mayor profusión sin mencionar uno de los lenguajes de más reciente aparición y que más ha evolucionado en los últimos años, considerando tanto sus características, como el número de herramientas que han surgido para su desarrollo e integración con otros lenguajes. Además, es el lenguaje que ha experimentado el mayor auge en los últimos años (sólo C# se puede comparar a él en este aspecto) en cuanto a utilización (está entre los seis o siete lenguajes más utilizados en el desarrollo de *software*). Nos estamos refiriendo a Python.

Python es un lenguaje de programación interpretado creado por Guido van Rossum en el año 1991. El nombre del lenguaje proviene de la afición de su creador original por el grupo de humoristas británicos *Monty Python* (`http://es.wikipedia.org/wiki/Monty_Python`). El principal objetivo que persigue este lenguaje es la facilidad, tanto de lectura, como de diseño. En la actualidad, Python se desarrolla como un proyecto de código abierto, administrado por la *Python Software Foundation*. Aunque se le considera como un lenguaje orientado a objetos, en su diseño asume también varios de los principios de la programación estructurada y algunos "toques" de la programación funcional. La lista es la estructura básica que permite agrupar colecciones de objetos en Python.

Python se utiliza como lenguaje de programación interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. Además, el intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo de un programa. Aunque ciertamente, los programas en Python tardan más en ejecutarse y consumen más recursos que sus equivalentes en lenguajes compilados (como C), tanto su curva de aprendizaje como su ciclo de desarrollo son bastante más cortos, lo cual le confiere una ventaja indudable. Además,

en la actualidad, el intérprete de Python es muy eficiente (ejecuta los programas más rápido y optimiza el consumo de memoria) e incluso existen herramientas que permiten compilarlo a una especie de *bytecode* propio más eficiente aún.

Si a estas características le añadimos el hecho de que es un lenguaje muy "limpio" y elegante para programar, podemos concluir que es uno de los lenguajes ideales con el que aprender programación, ya que permite centrarse rápidamente en los conceptos básicos de la programación estructurada para pasar, posteriormente, a la orientación a objetos sin necesidad de aprender otro lenguaje de programación.

Python, al igual que otros lenguajes de programación, permite dividir el programa en módulos reutilizables desde otros programas Python. Y lo que es más importante, desde el punto de vista del programador profesional, incluye una gran colección de módulos estándar que se pueden utilizar como base de los programas que desarrollemos (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan, entre otras, funciones para operar con ficheros, manipular sonido, crear interfaces gráficos de usuario y acceso a redes.

Fuera del paradigma de la POO, ha habido un lenguaje, difícilmente encasillable, que ha sido muy popular desde mediados de los noventa hasta ahora (aunque está empezando a perder bastantes adeptos debido al auge de otros lenguajes como Python y C#). Se trata de Perl, al que podríamos definir como lenguaje de *scripting*. El *scripting* es una técnica que consiste en desarrollar programas pequeños (desde pocas decenas hasta cientos de líneas de código) con propósitos muy específicos, generalmente relacionados con el proceso de grandes volúmenes de información en forma de texto con el fin de sacar estadísticas o extraer datos en base a la búsqueda de ciertos patrones.

Perl es un lenguaje de programación que fue diseñado por Larry Wall en 1987. Incorpora características del lenguaje de programación C, y también de ciertas herramientas presentes en los sistemas Unix y, en un grado inferior, de otros lenguajes de programación.

Estructuralmente, Perl está basado en el estilo de bloques utilizado por C y otros lenguajes de *scripting*, tal como AWK. Las principales razones de su éxito entre los programadores fueron su destreza en el procesamiento de grandes bloques texto y el hecho de no tener ninguna de las limitaciones de los otros lenguajes de *scripting*.

A mediados de los noventa, la difusión universal de Internet trajo consigo el desarrollo de nuevas funcionalidades para satisfacer la demanda existente de ejecución de programas de forma remota a través del WWW. Comenzaba a dar sus primeros pasos un nuevo paradigma de desarrollo de *software* basado en la ejecución de programas almacenados de forma remota en un servidor mediante la utilización de un navegador web.

En ese momento, no sólo surgieron nuevos lenguajes de programación basados en este concepto de desarrollo, como JavaScript o PHP, sino que también lenguajes ya existentes, como Java, se adaptaron a este nuevo modelo. Queremos resaltar que, pese a que sus nombres indiquen lo contrario, Java y JavaScript no tienen nada que ver entre sí. Contrariamente a lo que algunas personas creen, JavaScript no es ningún derivado de Java, sino más bien un lenguaje de *scripting* orientado al web y con características diferentes a las de Java.

En la actualidad, el modelo de desarrollo de aplicaciones y servicios vía web ha experimentado un auge considerable. Muchas de las aplicaciones informáticas que se

desarrollan hoy en día se almacenan en servidores y se ejecutan a través de Internet mediante el uso de navegadores web. Esto ha traído consigo la aparición de nuevas tecnologías y entornos de desarrollo orientados a la creación de programas bajo este modelo, como AJAX (*Asynchronous JavaScript And XML*) y .NET de Microsoft.

Precisamente, la creación de la plataforma .NET por parte de Microsoft trajo consigo el nacimiento de un lenguaje de programación muy usado en la actualidad: el C#, que es un lenguaje de programación orientado a objetos que incorpora algunas de las características de C, C++ y Java, aparte de otras nuevas funcionalidades que le confieren un enorme atractivo. Pero la plataforma .NET también permite que puedan desarrollarse programas empleando otros lenguajes, como, por ejemplo, Visual Basic. Precisamente, esto ha contribuido a darle un nuevo impulso a Visual Basic, siendo ahora otro de los lenguajes con un uso más extendido.

Plataformas de desarrollo como .NET tienen su origen en algunas herramientas que se desarrollaron en los años noventa. La idea básica consistía en disponer de un entorno profesional de creación de programas con el fin de aumentar el rendimiento a la hora de escribirlos: edición, compleción de código, resaltado de la sintaxis, depuración, herramientas para el desarrollo de interfaces gráficas, programación orientada a eventos (cuando hacemos *clic* con el ratón en un botón de una ventana se genera un evento que "captura" el programa, ejecutando, de esta manera, el código asociado), etc. Un ejemplo típico sería el entorno de desarrollo *Microsoft Visual Studio*.

A este tipo de herramientas, que incluían, obviamente, un lenguaje de programación, se les bautizó como la *cuarta generación* de lenguajes de programación. Se trataba de herramientas que permitían construir aplicaciones sencillas combinando piezas prefabricadas. Hoy se piensa que estas herramientas no son, propiamente hablando, lenguajes. Algunos proponen reservar el nombre de cuarta generación para la programación orientada a objetos. Estos lenguajes tienen una estructura similar a la de los lenguajes de alto nivel e incorporan características como:

- Acceso a base de datos corporativas (Oracle, MySQL, PostgreSQL...).
- Desarrollo de interfaces gráficas de usuario.
- Generación semi-automática de código.
- Programación "visual". Podemos enlazar instrucciones del programa a ciertos objetos de la interfaz gráfica (como, por ejemplo, botones en una ventana) y ver cómo se comportan al ejecutar el código.

También se habla de una *quinta generación* de lenguajes de programación, aunque realmente no existe un acuerdo al respecto. Se llama así, a veces, a los lenguajes propios de la inteligencia artificial, aunque con el fracaso del proyecto japonés para la construcción de ordenadores (*hardware*) de quinta generación el nombre ha caído en desuso.

La quinta generación de computadores fue un proyecto ambicioso lanzado por Japón a finales de los setenta. Su objetivo era el desarrollo de una clase de computadores que utilizarían técnicas de inteligencia artificial al nivel del lenguaje de máquina y serían capaces de resolver problemas complejos, como la traducción automática de

una lengua natural a otra (del japonés al inglés, por ejemplo). En los años noventa ya se pudo afirmar que el proyecto había fracasado.

El concepto de inteligencia artificial (IA) define la rama de la ciencia informática dedicada al desarrollo de agentes racionales no vivos (programas autónomos), entendiendo por agente a cualquier cosa capaz de percibir su entorno, procesar esas percepciones y actuar en él en función de las mismas. Por racionalidad se entiende la característica que posee una elección de ser correcta, más específicamente, de tender a maximizar un resultado esperado. Los lenguajes de programación, en el marco de la inteligencia artificial, se enmarcan en muchos ámbitos, como pueden ser el razonamiento basado en casos, los sistemas expertos, los sistemas multiagentes o el desarrollo de mundos virtuales y videojuegos.

Uno de los lenguajes más utilizados para la programación de sistemas multiagentes es JADE. La plataforma JADE (*Java Agent Development Framework*) se utiliza para el desarrollo de sistemas multiagentes. Desde una perspectiva general, JADE es un entorno de trabajo para el desarrollo de *software*, orientado al despliegue de sistemas multiagentes y aplicaciones relacionadas con los estándares FIPA (*Foundation for Intelligent Physical Agents*) para agentes inteligentes. La plataforma JADE no sólo proporciona el entorno donde los agentes "viven", sino que también proporciona un conjunto de librerías para el desarrollo de agentes (utilizando como soporte el lenguaje JAVA), que puedan comunicarse entre sí, haciendo uso del lenguaje de comunicación FIPA.

En el campo de las aplicaciones orientadas a la informática gráfica (sobre todo en 3D), destaca OpenGL (*Open Graphics Library*). OpenGL es una especificación estándar que define una colección de funcionalidades multiplataforma (que puede ser empleada por varios lenguajes de programación sobre tipos de ordenadores diferentes) para escribir aplicaciones que produzcan gráficos 2D y 3D. OpenGL se utiliza, sobre todo, en programas desarrollados con C++ para el desarrollo de entornos virtuales y videojuegos. La interfaz consiste en más de 250 funciones diferentes que pueden usarse para dibujar escenas tridimensionales complejas a partir de primitivas geométricas simples, tales como puntos, líneas y triángulos. Fue desarrollada originalmente por Silicon Graphics Inc. (SGI) en 1992 y se usa ampliamente en CAD (*Computer Aided Design*, Diseño Asistido por Ordenador, en castellano), realidad virtual, representación científica, visualización de información y simulación de vuelo.

Otro tipo de lenguajes aplicados en la IA son los lenguajes para el desarrollo de redes neuronales. Las redes neuronales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales. Un ejemplo de aplicación de redes neuronales es el desarrollo de *bots*. Un *bot* (diminutivo de robot) es un programa informático que realiza funciones muy diversas, imitando el comportamiento de un humano. Un ejemplo de *bot* es el *Neuralbot*, un bot para el juego Quake II que utiliza una red neuronal para decidir su comportamiento y un algoritmo genético para su aprendizaje.

Otros campos atractivos de la investigación informática, derivados inicialmente de la IA, pero con entidad propia desde hace tiempo, son los del Reconocimiento de Formas (RF) y la Visión por Computador (VC).

En el campo del Reconocimiento de Formas se abordan problemas relacionados, en cierta forma, con la manera que tenemos los seres humanos de reconocer la infor-

mación (datos en general) que tenemos en nuestro entorno. Así, por ejemplo, algunos de los problemas que aborda el RF son la construcción de sistemas de reconocimiento automático del habla (RAH), de sistemas de reconocimiento óptico de caracteres (OCR, por *Optical Character Recognition*), de sistemas de ayuda al reconocimiento de escritura manuscrita, o de sistemas para la identificación de individuos a partir de alguna de sus características físicas, tales como la huella dactilar, el tono de voz, o el iris de los ojos (biometría).

El campo de investigación de la Visión por Computador intenta resolver problemas relacionados con el procesamiento de imágenes en general. Así, la VC busca desarrollar sistemas que permitan analizar escenarios (por ejemplo, para identificar objetos determinados en una imagen), o sistemas que permitan el reconocimiento facial, es decir, identificar un rostro en una escena, probablemente en movimiento.

No se puede afirmar con rotundidad que haya una serie de lenguajes de programación "propios" del RF o de la VC, sino que, más bien, se utilizan las herramientas de desarrollo que mayor comodidad ofrecen para los programadores en general. Resulta bastante común el empleo de lenguajes de programación orientados a objetos, como C++, Java o Python, e incluso plataformas de desarrollo que facilitan la creación de interfaces gráficas de usuario y tienen funcionalidades que permiten operar con *hardware* específico para la adquisición de datos (pantallas táctiles, lápices ópticos, etc.), tales como .NET. En este último caso se suele emplear el lenguaje de programación C#, aunque .NET también permite integrar código desarrollado en Python y en otros lenguajes de programación.

Por último, podemos afirmar que ambos campos guardan cierta relación con la *informática industrial*, ya que, en muchas situaciones, se desea conectar dispositivos específicos de captura de datos (cámaras, por ejemplo) al sistema operativo (SO) del ordenador a través de circuitos integrados (tarjetas). En esos casos, puede ser necesario desarrollar un *driver* o programa que pueda integrarse al SO para controlar dicho dispositivo e interactuar con el sistema de RF o CV correspondiente. A la hora de desarrollar este tipo de programas específicos de bajo nivel se usa, en la mayoría de las ocasiones, el lenguaje de programación C, combinándolo, algunas veces, con instrucciones en lenguaje ensamblador.

Pero no sólo se dan estas situaciones con dispositivos de captura de datos. También puede ocurrir cuando se desarrollan sistemas específicos de control. Piénsese, por ejemplo, en la utilización de un sistema de control de la cabina de una ascensor mediante la voz: es necesario diseñar un circuito integrado que permita controlar los mecanismos del ascensor a través del sistema operativo del ordenador y que interactúe mediante un *driver* específico con el sistema de reconocimiento de voz propiamente dicho.

CAPÍTULO 2

SISTEMAS OPERATIVOS Y ENTORNOS DE USUARIO

2.1. EL SISTEMA OPERATIVO

En el capítulo anterior aprendimos que no existe un único modelo de ordenador, sino que, en realidad, existen varias *arquitecturas*. Incluso dentro de una misma arquitectura podemos encontrarnos con modelos posiblemente muy diferentes entre sí, por ejemplo un ordenador portátil es muy distinto de un ordenador de sobremesa, aunque ambos sean PC.

¿Es necesario que tengamos que aprender a manejar y/o programar cada ordenador como si fuese totalmente diferente a los demás? Esto haría muy ineficiente el manejo de ordenadores y, desde luego, imposibilitaría su manipulación a usuarios que no tuviesen conocimientos altamente técnicos.





Figura 2.1. Diferentes tipos de ordenadores: un PC de sobremesa y un portátil

Para solucionar los problemas derivados de la heterogeneidad de los ordenadores existe un programa denominado sistema operativo (S.O.), que se encarga de "manejar" el ordenador por nosotros. Como vimos en el capítulo anterior, el ordenador controla los dispositivos y recibe nuestras órdenes para interactuar con ellos. Por tanto, el S.O. es un programa que proporciona una *visión homogénea* de computadores posiblemente (muy) distintos entre sí. En este sentido:

- Proporciona abstracciones del hardware que lo hacen independiente de los detalles específicos de su funcionamiento. Por ejemplo, tanto los discos duros como

los CD-ROM muestran su contenido como una jerarquía de directorios que contienen ficheros y, a su vez, otros directorios con ficheros y así sucesivamente.

- Facilita al programador un repertorio de funcionalidades idénticas que le permiten acceder a dispositivos diferentes de la misma forma. Por ejemplo, el programador usa las mismas instrucciones para escribir datos en un disco duro o un disquete.

Sin embargo, no existe un único S.O. que sea comúnmente utilizado por todos los usuarios de ordenadores. En el "mercado" podemos encontrar varios, cada uno con sus ventajas e inconvenientes. Así el profesional de la informática y/o administrador de sistemas, normalmente, trabajará con sistemas operativos de la "familia" Unix[1]. El profesional del diseño gráfico probablemente trabaje con OS X en un Apple Macintosh. En el ordenador de casa probablemente se utilice un S.O. de la "familia" Windows, más apropiado para juegos, ofimática y aplicaciones multimedia[2]. ¿Qué podemos encontrar?

- Los de la familia **Microsoft Windows**: Windows 98, Windows Me, Windows NT, Windows 2003, Windows XP y Windows Vista. Son sistemas orientados a usuarios domésticos y ofimática. Windows 98, Me y NT están obsoletos y ya no tienen soporte de Microsoft, ni tan siquiera actualizaciones de seguridad. Windows 2003 y XP tienen versiones profesionales más orientadas a servidores y programación de aplicaciones.

- Los de la familia **Unix**: Unix, BSD, FreeBSD, Linux, Solaris, AIX, HP-UX. . . , orientados a servidores y programación. En los últimos años, han aparecido entornos y herramientas de trabajo que lo acercan también a usuarios domésticos y ofimática (Linux, en particular).

- Los de la familia **Apple Macintosh**: OS 9, OS X (*a la Unix*). Orientados a usuarios domésticos, ofimática y diseño gráfico. La nueva versión, OS X, es un derivado de Unix. Destacan por su facilidad de uso, pero *sólo funcionan sobre ordenadores de la marca Apple*.

2.1.1. Clasificación de los sistemas operativos

Los sistemas operativos se pueden clasificar, desde el punto de vista de la ejecución de procesos, en:

- *Monotarea*: el primer proceso que entra en ejecución monopoliza la CPU hasta su fin, es decir sólo permiten ejecutar un programa cada vez. Esto resulta muy sencillo de gestionar para el S.O. Ejemplos: MS-DOS, Windows 3.X.

[1]Con la llegada de Windows XP Profesional y la "saga" 2000 Server (el último es el Windows 2003 Server) ya se empieza a disponer de servidores medianamente fiables equipados con productos de Microsoft.

[2]Linux ha evolucionado muchísimo en lo que se refiere al funcionamiento de sus aplicaciones ofimáticas y capacidad multimedia. Hoy en día, resulta discutible decir que Windows es más apropiado para estos campos.

- *Multitarea*: todos los procesos que se ejecutan comparten la CPU, es decir varios programas se ejecutan "simultáneamente"[3]. Los sistemas modernos son multitarea. Exige un buen *algoritmo de planificación* y una adecuada gestión de procesos por parte del S.O. En la práctica hay un límite al número de procesos, que viene marcado por la estructura interna del S.O. y la memoria principal del ordenador. Ejemplos: Linux, Windows 9X.

Desde el punto de vista del número de usuarios que están autorizados a trabajar con el ordenador al mismo tiempo, distinguimos:

- *Monousuario*: solamente un usuario puede estar trabajando con la máquina en un momento dado. No requiere que el sistema de ficheros adopte un mecanismo de protección de datos (permisos) entre usuarios, ni se evita que el usuario pueda reconfigurar el ordenador o destruir ciertos archivos del sistema. Ejemplos: Windows 9X.
- *Multiusuario*: hay varios usuarios autorizados a conectarse a la máquina. Requiere mecanismos seguros de identificación (identificación y clave) y de protección en el sistema de ficheros. Igualmente garantiza que las acciones de un usuario cualquiera no interfiere con las de otros (un usuario no puede, por ejemplo, destruir archivos de otro, a no ser que se le dé permiso para ello). Ejemplos: Linux, Windows 2003, Windows XP...

2.1.2. Sistemas multiusuario

La UJI utiliza un sistema UNIX (Linux) en el ordenador que hace de servidor y almacena las "cuentas" (espacio en disco) de todos sus alumnos. UNIX es un sistema multiusuario. Esto hace que sea necesario establecer mecanismos para identificar al usuario y para proteger unos usuarios de otros. Cada usuario de la máquina tiene un UID (user id) o *nombre de usuario* y puede pertenecer a *uno o más grupos* identificados por su GID (group id). Cada grupo reúne a usuarios con determinado perfil. Por ejemplo: profesores, alumnos, servicios, etc. Uno de los grupos a los que pertenece el usuario es el que se toma como *grupo propietario*. Son el UID y el GID del usuario que lanza un proceso los que determinan qué puede o no hacer ese proceso. Cada usuario que se da de alta en el sistema suele tener un espacio en el disco duro para almacenar sus ficheros (normalmente limitado).

¿Sabías que...?

En realidad, se puede hacer que el proceso tenga un UID y un GID distintos de los del usuario que los lanza. Esto puede tener importantes consecuencias de seguridad.

[3]En realidad, se ejecutan poco a poco por turnos. En cada turno un único proceso monopoliza la CPU, pero como esto ocurre a intervalos de tiempo tan pequeños (del orden de milisegundos) se crea la ilusión de simultaneidad.

Para iniciar una sesión de trabajo, el usuario debe identificarse ante el sistema. Si lo hace correctamente, tiene acceso a determinados recursos y puede ejecutar ciertas acciones en función de los grupos a los que está adscrito. Para identificarse, los usuarios introducen su nombre de usuario (el login) y la correspondiente palabra de paso (*password*) o *contraseña*.

Cuando creas una nueva cuenta (un nuevo usuario) como administrador, debes introducir una contraseña para el usuario. La Universidad te ha asignado ya una contraseña para acceder a un ordenador de alumnos de la UJI (`anubis`) que proporciona servicios de correo electrónico y te ofrece una cuenta (espacio en disco protegido con identificador de usuario y contraseña) para prácticas. Has de cambiar la contraseña cada cierto tiempo. Cada vez que te conectes a cualquier ordenador de las aulas de prácticas (Linux) usarás el usuario y contraseña que se te facilitaron al matricularte en la UJI. Al conectarte, accederás a tu espacio en disco de `anubis` mediante la red de la UJI. Aunque el servidor de alumnos es `anubis.uji.es`, en realidad se emplea el sistema "gemelo" `lynx.uji.es` para el acceso interactivo.

Para satisfacer tu curiosidad...

Las palabras de paso de todos los usuarios están, junto con otras informaciones, en el fichero `/etc/passwd`, que puede ser leído por cualquiera.

Para evitar problemas de seguridad, los *passwords* se guardan codificados mediante una función que es relativamente fácil de calcular, pero cuya inversa es muy difícil de encontrar.

Para verificar el password introducido por el usuario, primero se codifica y después se compara con la versión guardada.

Has de tener cuidado a la hora de elegir la contraseña que permite el acceso a tu cuenta. Si eliges una muy "fácil" cualquiera podría entrar en el sistema haciéndose pasar por ti. Algunas indicaciones sobre cómo elegir una buena contraseña.

- No hay que elegir palabras comunes (presentes en un diccionario) ni nombres, especialmente el propio, de familiares o de personajes famosos.
- Mezclar letras y números. No utilizar *passwords* que sean totalmente numéricos y nunca el número de teléfono, DNI o similares.
- Elegir contraseñas de seis a ocho caracteres (o más si es posible).
- Utilizar contraseñas distintas para máquinas distintas.
- Hay que ser cuidadoso con los caracteres especiales, pues pueden ser representados "internamente" de formas distintas según la configuración del sistema (véase el capítulo siguiente sobre codificación de la información en los ordenadores). *Es mejor que emplees sólo letras y dígitos*.

- Una idea sencilla es pensar un nombre relacionado con alguna afición o actividad que nos guste (p.e. mitología romana) y después *cambiar* todas las letras *i* por 1, las letras *o* por 0, etc. La palabra resultante la usaríamos como *password*. Por ejemplo: Mercurio (el mensajero de los dioses) da como resultado la contraseña mercur10 (no uséis esta).
- No escribáis nunca la contraseña en un papel.

Pros y contras del `/etc/passwd`

Al guardarse las palabras de paso codificadas, se evita el riesgo de que alguien las encuentre "accidentalmente".

Al tener permisos de lectura para todo el mundo, permite mayor flexibilidad a los programas (p.e. un programa salvapantallas puede preguntar de nuevo la palabra de paso al usuario, para asegurarse de que no se ha ido a tomarse café y otro ha ocupado su lugar).

Sin embargo, si las palabras de paso elegidas son poco seguras, es un sistema vulnerable ante el "ataque con diccionario": se compara el fichero con una lista de palabras comunes ya codificadas.

Para solucionarlo se han introducido las llamadas *shadow password*. La idea es guardar las palabras de paso en un fichero separado y protegido contra lectura.

En los sistemas de la "familia" Unix, existe un número de UID especial, el 0. Éste corresponde a un usuario especial: el *superusuario*, que generalmente recibe el nombre de *root*. Ese usuario puede "saltarse" los permisos de los ficheros; puede leer y escribir (es decir, modificar o borrar) cualquier fichero. Es el encargado de la (buena) gestión de la máquina a su cargo. Tiene poder para detener cualquier proceso del sistema. Es importante que su contraseña no sea conocida más que por la persona a cargo del sistema.

Por tanto, debes ir con cuidado con la información que consideres privada o confidencial, ya que el superusuario siempre puede acceder a ella. Aunque habría que resaltar que, *por cuestiones de ética profesional*, el superusuario *no debe* violar la confidencialidad de los datos de los usuarios[4]; más bien al contrario. Cualquier actitud abusiva al respecto podría ser incluso objeto de denuncia.

2.1.3. Plataformas

Como vimos en el capítulo anterior, el S.O. se encarga de la gestión de procesos, siendo, por tanto, el responsable de la ejecución de los programas. La ejecución de programas está, pues, íntimamente ligada al S.O. De hecho el S.O. puede ejecutar los

[4]Salvo por imperativos legales o razones de "peso" que tengan que ver con la correcta administración del sistema a su cargo.

programas porque en los ficheros que los contienen existe información que le indica cómo hacerlo. Esa información es particular para cada S.O.

Esto quiere decir que los programas se compilan también para ser ejecutados en un S.O. concreto. Sin embargo, también en el capítulo anterior, habíamos visto que los programas se compilaban para una *arquitectura* específica. ¿Qué quiere decir esto?

Ambas cosas son ciertas. Los programas se compilan para un S.O. concreto que se ejecuta en una arquitectura específica (tipo de ordenador). El conjunto arquitectura + S.O. conforma lo que comúnmente se denomina *plataforma*. De esta manera los programas se compilan para ser ejecutados en una plataforma determinada. Así, por ejemplo, un programa compilado para PC bajo Windows no funcionará en un PC con Linux y viceversa. Necesitamos coger el código fuente del programa y compilarlo en la plataforma correspondiente[5].

Cada sistema operativo se acompaña, por tanto, de un conjunto de utilidades característico (programas compilados para ese S.O. y el tipo de ordenador sobre el que van a funcionar). Así, en muchos Unix encontrarás la misma colección de utilidades para gestionar ficheros, editar texto, etc. El sistema operativo Linux, por ejemplo, se acompaña de una serie de utilidades construidas por GNU (un grupo de programación de software libre), así que hablamos del sistema GNU/Linux. Estas utilidades han sido compiladas para las distintas arquitecturas en las que se puede instalar Linux (x86, Alpha, Sparc. . .).

2.1.4. Entornos de usuario

Cada sistema operativo ofrece uno o más *entornos de usuario* (o entornos de trabajo). El entorno de usuario proporciona herramientas para acceder a la información, ejecutar aplicaciones, configurar el ordenador, etc. Los entornos de usuario son la interfaz que sirve para establecer la "comunicación" entre el usuario y el S.O. Tipos de entornos que podemos encontrarnos:

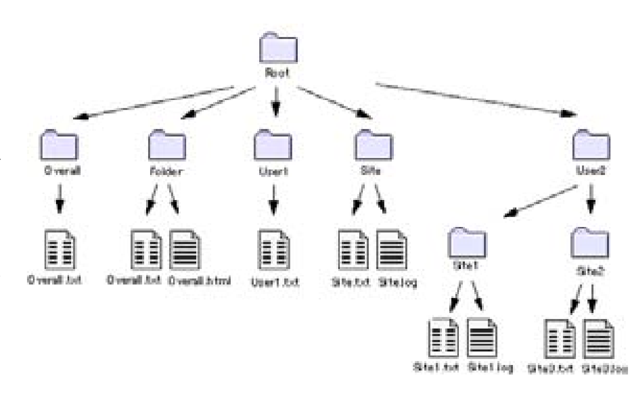


Figura 2.2. Organización jerárquica de un sistema de ficheros

- De *línea de órdenes*: las órdenes se expresan como texto.

  Ejemplo: Los sistemas Unix proporcionan un rico juego de órdenes y uno o más intérpretes de órdenes o *shells* (programas que leen órdenes de teclado y controlan la ejecución de las mismas).

- De *interfaz gráfico*: las órdenes se expresan accionando sobre iconos visuales, botones y menús con un ratón.

[5]Puede que incluso tengamos que hacer alguna modificación en el código original.

Ejemplos: Microsoft Windows y Apple Macintosh presentan sendos entornos de interfaz gráfico. Linux permite elegir entre varios entornos de usuario (aunque hay dos más avanzados y populares: Gnome y KDE).

Acerca de los intérpretes de órdenes

La interacción de los usuarios con el sistema operativo se hace a través de los intérpretes de órdenes (*shells*).

Como veremos en el capítulo 6, las órdenes suelen ser poco amigables (modo texto en consola). Se suele intentar evitar la redundancia y hacer órdenes especializadas. En la mayoría de los casos, el éxito se traduce en el más absoluto silencio.

Los *shells* incluyen sus propios "lenguajes de programación" que, además, suelen ser muy potentes. Podemos elegir entre diversas opciones: por ejemplo, `/bin/sh`, `/bin/ksh`, `/bin/csh`, `/bin/bash`. El superusuario asigna el programa intérprete de órdenes en el momento de dar de alta al usuario en el sistema.

Es importante que observes que dos ordenadores diferentes, con un mismo entorno de usuario, se manejan del mismo modo. Por ejemplo, el entorno KDE funciona igual ejecutado en un PC que en una estación de trabajo SUN. Antes de estudiar los entornos de usuario, hemos de introducir conceptos como el de sistema de ficheros.

Figura 2.3. Visión del sistema de ficheros como un archivador

2.2. SISTEMAS DE FICHEROS

El sistema de ficheros es la parte del S.O. que se encarga de la representación *abstracta* de los dispositivos de almacenamiento secundario que pueden conectarse a un ordenador: discos duros, otros discos externos o internos (p.e. SCSI o USB), disquetes, CD-ROM, DVD, etc. Desde este punto de vista el S.O. se encarga, fundamentalmente, de:

- La gestión de los ficheros, esto es, de la correspondencia entre colecciones de información relacionadas entre sí y agrupadas bajo un nombre único dado por su creador (una imagen, una pista de música, un documento, una hoja de cálculo, etc.) con sectores físicos.
- La organización *jerárquica* de los ficheros agrupados en colecciones de carpetas o directorios, posiblemente distribuidos en dispositivos diferentes.

Por consiguiente, el sistema de ficheros proporciona una visión de la información almacenada en unidades de almacenamiento como *ficheros agrupados en una jerar-*

quía de directorios. Se pretende establecer una analogía con el archivador de una oficina.

- El ordenador es como un gran archivador.
- Cada dispositivo o unidad (disco duro, disquete, CD-ROM, etc.) es un cajón del archivador.
- Cada directorio es una carpeta. Dentro de una carpeta podemos almacenar otras carpetas y, del mismo modo, un directorio puede contener otros directorios.
- Cada fichero es un documento.

¿Sabías que…?

Para poder llevar a cabo eficazmente las tareas de gestión del sistema de ficheros, el S.O. define internamente ciertas *unidades lógicas* de información que permiten realizar una representación básica de las *unidades físicas* que conforman los dispositivos

Por ejemplo, los sectores en que se divide la cinta magnética del disco duro y que sirven para organizar la información en él almacenada.

La operación consistente en determinar la cantidad y la representación inicial de las unidades lógicas necesarias para establecer la correspondencia con las unidades físicas de un dispositivo y almacenarlas en el mismo se denomina comúnmente *formatear* o *dar formato*

Formatear supone la "construcción" del sistema de ficheros en el dispositivo. Esta operación normalmente destruye los datos que pudiera haber almacenados en la unidad.

Cualquier unidad de almacenamiento secundario que queramos conectar a nuestro ordenador y a la que deseemos acceder mediante las herramientas proporcionadas por el sistema de ficheros de un determinado S.O. habrá de estar *formateada* de manera que pueda ser accesible para dicho S.O.

Pensarás que hay excepciones a esto que hemos dicho: por ejemplo el CD-ROM. Ten en cuenta, no obstante, que, como veremos más adelante, los CD-ROM vienen ya "formateados" con un sistema de ficheros específico que se crea en el momento de grabarlos y que "entienden" y reconocen todos los sistemas operativos.

En los entornos gráficos de usuario los *ficheros* se muestran con diferentes iconos, en función del tipo de contenido. Distinguimos, fundamentalmente, dos tipos de fichero:

- *Aplicaciones* o programas: son ficheros ejecutables.



- *Documentos*.



Los *directorios* se muestran como carpetas:



Las *unidades* o dispositivos conectados al ordenador se representan en Windows con iconos especiales. Linux puede utilizar indistintamente iconos especiales o carpetas para representar los dispositivos.



El *ordenador* se representa con un icono que representa un PC (Windows) o como una carpeta (Linux):

2.2.1. Ficheros y directorios

Las características o propiedades que tiene un fichero son:

- *Nombre*: secuencia de caracteres que lo identifica en un directorio.

  Se suele descomponer en dos partes:

  - *Extensión*: los caracteres que siguen al último punto del nombre. Se usan para indicar el tipo del archivo. El icono del fichero se suele determinar a partir de la extensión. ¡No es obligado utilizar una extensión! Sí es cierto que se acostumbra a usarla y que muchos entornos gráficos la utilizan para representar el fichero con un determinado icono e, incluso, para "asociar" ficheros con programas (los programas que se ejecutarán al abrir el fichero).
  - Y el *nombre* propiamente dicho.

  Ejemplo: `fichero.txt` es el nombre de un archivo con extensión `txt`, que indica que es un fichero de texto. Ten en cuenta que en sistemas antiguos como MS-DOS se imponían ciertas restricciones a los nombres de archivo: el nombre propiamente dicho no podía superar ocho caracteres y la extensión, tres.

- *Atributos*: una serie de valores que indican cosas tales como:

  Ubicación, dónde está situado en el disco (*path* o ruta de acceso, como veremos posteriormente).

  Tamaño, el tamaño actual (en alguna unidad adecuada) y (posiblemente) el máximo tamaño permitido.

  Protección, información para controlar quién y cómo puede utilizar el fichero; esto es, los permisos de seguridad (o utilización) que indican si el fichero es legible, modificable, ejecutable, etc.

  Contador de uso, número de procesos (programas) que están usando el fichero.

  Hora y fecha de la última modificación efectuada.

  ...

- *Contenido*: la información que hay en su "interior" (texto, dibujos, música, etc.).

Las características o propiedades que tienen los *directorios* son:

- *Nombre*: normalmente sin extensión. Nuevamente, conviene indicar que ésta es una costumbre, *no una obligación*. Puedo usar un nombre con extensión para identificar un directorio.
- *Atributos*:
  - fecha de creación y fecha de la última modificación,
  - permisos de seguridad ("listable", modificable, con contenido accesible, etc.),
  - ...
- *Contenido*: ¿Qué contiene un directorio? Pues otros directorios y/o ficheros.

2.2.2. Acciones

Los ficheros son los elementos básicos para la organización lógica de la información en el sistema de ficheros. Las carpetas o directorios son un mecanismo para la organización jerárquica de la información[6]. Las carpetas contienen ficheros y pueden contener otras carpetas. Las operaciones básicas que podemos realizar para administrar carpetas y ficheros son:

- Crearlos.
- Copiarlos.
- Borrarlos.
- "Moverlos", esto es, cambiarlos de ubicación. Esto incluye también cambiarles el nombre (renombrarlos).

[6]Este tipo de organización es la más común.

- Comprimirlos.
- Modificar su contenido.
- Ver (listar) su contenido.
- Empaquetarlos.
- ...

La forma en que se realizan estas acciones suele ser bastante similar en la mayoría de los entornos gráficos de usuario. Por ejemplo: borrar un fichero en un entorno gráfico suele consistir en desplazar el icono que lo representa sobre el icono que representa una papelera[7]:

2.3. EL ESCRITORIO DE WINDOWS XP

Hace poco tiempo, Microsoft informó que la rama Windows 98 (95, 98, Me) ya no iba a ser mantenida y, de hecho, ya no tiene soporte. Eso implica falta de actualizaciones de seguridad, parches de mantenimiento, soporte para nuevo hardware, etc. Con ello, Microsoft está avisando a los usuarios de estos sistemas para que consideren seriamente la migración a otros sistemas Windows (con el correspondiente abono de una nueva licencia).

A ello se une el hecho de que, desde hace un par de años, la rama NT (todas las versiones de Windows NT) ya no tiene soporte oficial de Microsoft y su desarrollo ha sido abandonado por completo. La solución ofrecida consistió en migrar a la serie Windows 2000 Server (la última versión es Windows 2003 Server). La serie 2000 Server supone, de hecho, la "sucesión natural" de NT, ya que todas las versiones 2000 Server están basadas en la tecnología NT (mejor dicho, en una versión desarrollada basándose en la tecnología introducida en NT).

Recientemente, Microsoft decidió reunir la interfaz de usuario de Windows 98 (el escritorio y sus funcionalidades, la barra de tareas con sus iconos, el menú Inicio, Internet Explorer, la tecnología *Plug&Play*...) y la tecnología desarrollada a partir de NT (sistema de ficheros NTFS más avanzado, planificador de procesos, mejor gestión de fallos, capacidades multiusuario...) en un único S.O., *Windows XP* (véase la Figura 2.4), para el que se realizaron dos versiones:

1.- Una dedicada al profesional de la informática y administradores de sistemas: *Profesional*. Es la versión completa, incluyendo herramientas de administración, mayores sistemas de seguridad, capacidad para conectar más de cinco ordenadores en red y soporte para múltiples procesadores en una misma placa.

[7] Normalmente, después, hay que vaciar la papelera para borrarlo *definitivamente*. Mientras esté en la papelera y ésta no se vacíe, el fichero se puede *recuperar*.

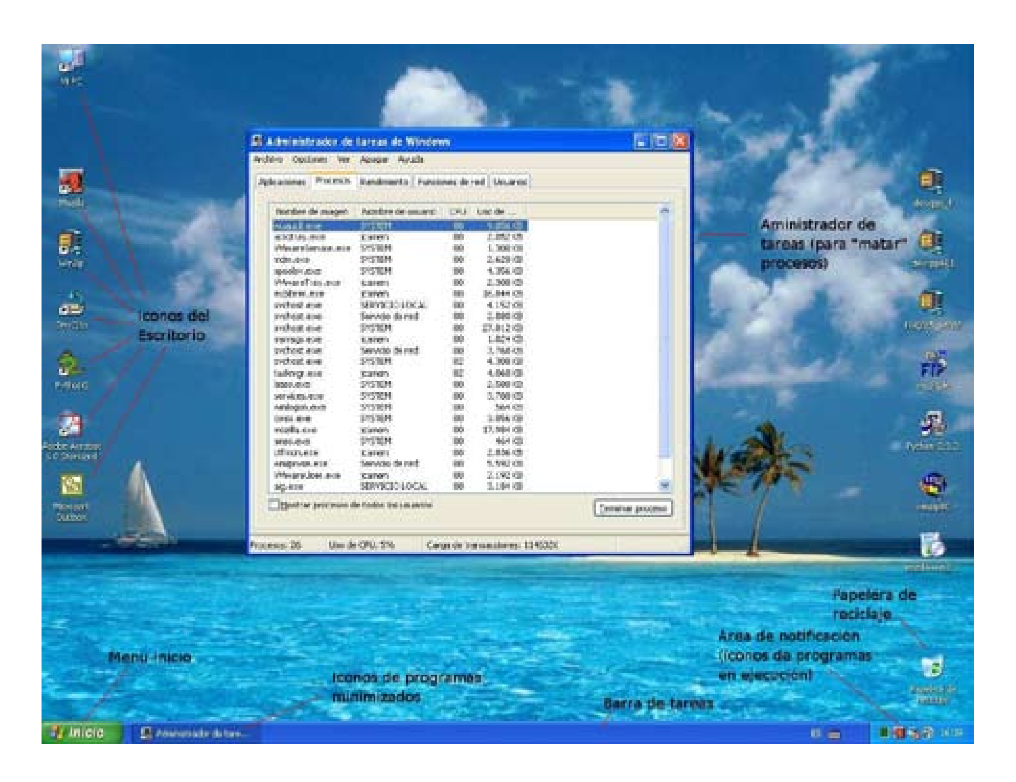


Figura 2.4. Una imagen del escritorio de Windows XP

2.- Otra dedicada a los usuarios caseros sin (o con escasos) conocimientos de informática[8]: *Home Edition*. Ésta es una *versión simplificada* que carece de las herramientas de administración, ofrece una menor seguridad que la versión profesional, sólo admite un máximo de cinco equipos conectados en red y no soporta más de un procesador en la placa (por tanto, no funciona con placas base duales).

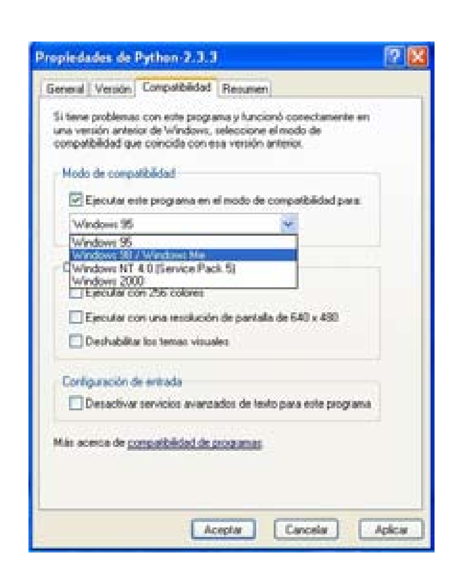


Figura 2.5. Activando modo de compatibilidad en XP

Windows XP ha partido del *kernel* (núcleo) de NT y lo ha mejorado un poco. Por consiguiente, desde el punto de vista de la gestión de procesos cabe destacar que *existe separación entre los mismos*, como, de hecho, existe en Unix desde hace muchos años. Esto significa que se pueden "matar" procesos de la misma manera que se puede hacer desde hace ya mucho tiempo en los sistemas Unix (incluyendo a Linux).

Ésta es una medida importante desde el punto de vista de la estabilidad del sistema: podemos terminar la ejecución de aquellos procesos que "cascan" o bien se tornan "inestables" y que antes (Windows 98) hacían que el sistema se "colgase" o bien que apareciesen las temibles "pantallas azules de la muerte". Si quieres acabar con algún programa en ejecución (proceso) pulsa la combinación de teclas CTRL+ALT+SUPR

[8]Los "meros mortales", para que nos entendamos ;-).

y aparecerá una ventana con la lista de todos los procesos activos. Selecciona el que desees parar y pulsa *Terminar proceso*.

Si estás habituado a usar Windows 98, no te resultará difícil adaptarte al escritorio de Windows XP. Su aspecto por defecto es diferente, pero la forma de funcionar del escritorio es análoga a Windows 98, ya que puedes gestionar tus carpetas y ficheros de modo similar. También podrás ejecutar programas de la misma manera, aunque para instalarlos necesitarás tener privilegios en el sistema (ser usuario administrador).



(a) Menú contextual para Inicio



(b) Ventana para seleccionar la vista clásica

Figura 2.6. Para cambiar el aspecto de Windows XP

Puede ser que algunos programas antiguos que funcionaban con Windows 98 no te funcionen ahora. En ese caso, prueba a ejecutarlos en el modo de compatibilidad con Windows 98 que se ha definido en XP para proporcionar un "entorno de ejecución" similar al de ese sistema. Observa la Figura 2.5.

Si quieres que el escritorio de XP presente un aspecto más parecido al clásico de Windows 98, basta con que hagas clic con el *botón derecho* del ratón sobre el botón que representa al *menú Inicio*, escojas las opción *Propiedades* en el menú contextual que se despliega, marques la opción *Menú Inicio clásico* de la ventana que emergerá y pulses *OK*. Observa atentamente la Figura 2.6.

En Windows XP ya no encontrarás los iconos de *Mi PC*, *Mis Documentos* e *Internet Explorer* en el Escritorio. Por defecto aparecen en el *menú Inicio*. Además, cada usuario tiene su propia carpeta *Mis Documentos* (recuerda que Windows XP es un sistema multiusuario).

Para trasladarlos al Escritorio basta con que selecciones (mediante un clic) la correspondiente entrada en el menú y, *sin soltar el botón del ratón*, la "arrastres" y la sueltes en cualquier lugar despejado del Escritorio.

Igualmente, en Windows XP puedes *administrar usuarios y grupos* (sólo la versión Profesional lo permite plenamente). La forma de gestionarlos es similar a como se hace en Unix. Como muestra la figura 2.7 puedes crear una nueva cuenta de usuario, modificar las características de una ya existente (cambiar privilegios, contraseñas...), cambiar las propiedades de la sesión de los usuarios, etc.

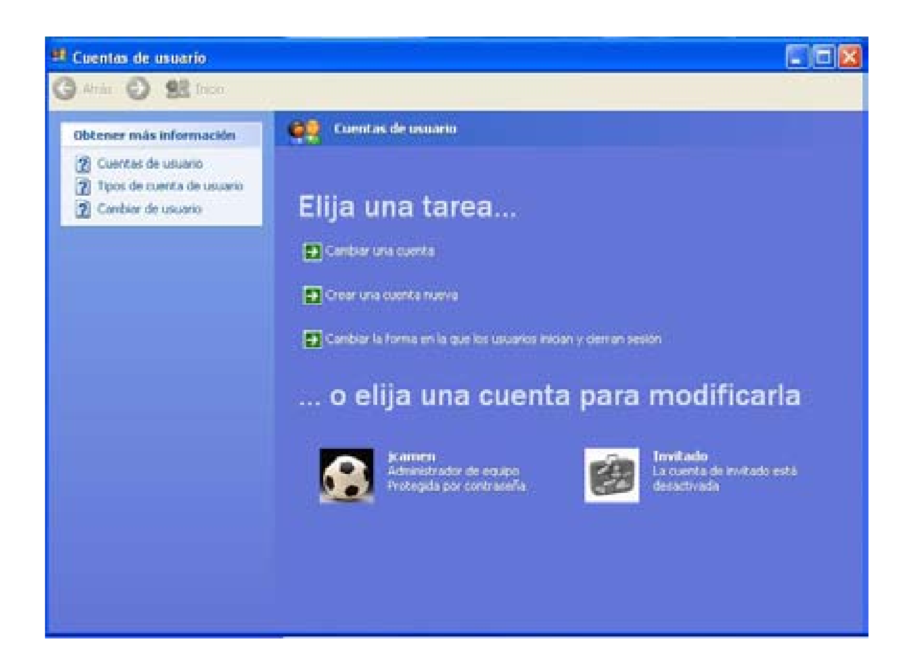


Figura 2.7. Las cuentas de usuario en Windows XP

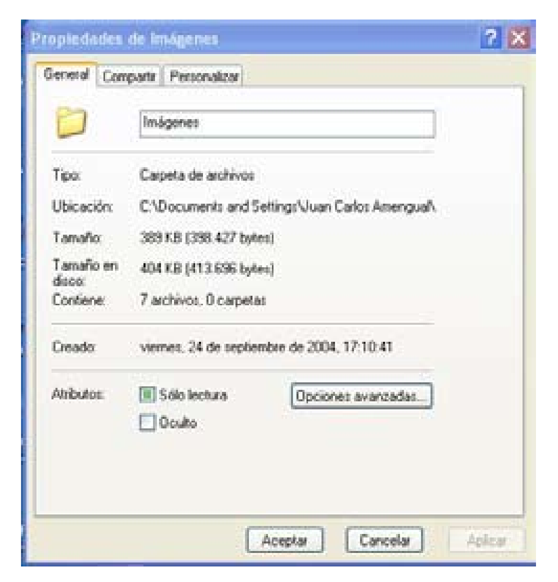
También puedes “trastear” con los permisos de ficheros y directorios. Si accedes a la opción *Propiedades* del menú contextual (emerge haciendo clic con el botón derecho del ratón sobre el icono que representa al objeto) de una carpeta o fichero, tal como muestra la figura 2.8, podrás:

- En la pestaña *General*: ocultar el objeto o bien fijar el atributo de “Sólo lectura” para que nadie pueda modificarlo.
- En la pestaña *Compartir*: compartir el objeto con otros usuarios del sistema o bien convertir el objeto en *privado* (sólo tú tienes acceso), compartir con otros usuarios de la red local, etc.
- En las *Opciones avanzadas* de la pestaña General: cifrar o comprimir los datos automáticamente y permitir que el sistema de búsqueda de Windows lo indexe para facilitar posteriores búsquedas de ficheros y carpetas en el PC.

2.4. EL ENTORNO DE TRABAJO UNIX

En las sesiones de prácticas utilizarás los entornos gráficos (los “escritorios”) de Microsoft Windows y Linux. Ya hemos hablado brevemente de Windows XP en la sección anterior. En el caso de Linux solamente estudiaremos el entorno gráfico de usuario KDE (K Desktop Environment). Existen otros, como GNOME.

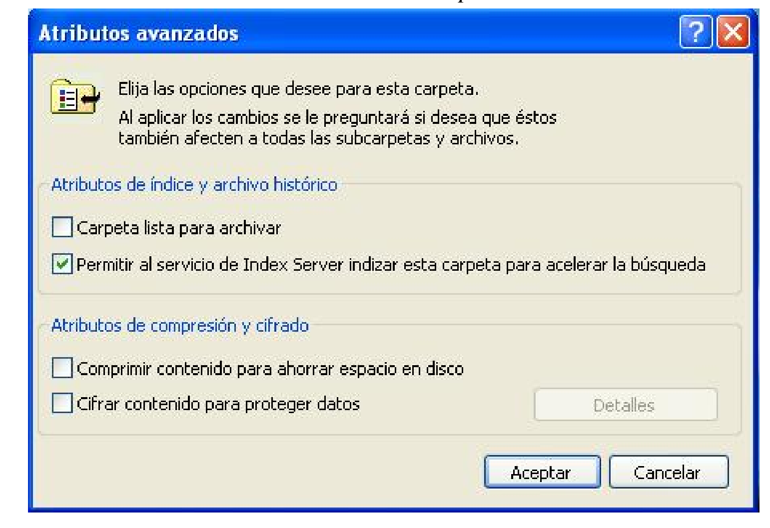
En las clases de teoría haremos una breve introducción a KDE, mostrando mediante ejemplos la realización de cada una de las acciones básicas en dicho entorno gráfico. No lo hacemos por la dificultad de manejo, sino más bien por la novedad que supone para la mayoría de vosotros. Ten en cuenta que KDE es un entorno tan sencillo de usar como el de Windows y presenta características muy similares en cuanto a funciona-



(a) Propiedades de la carpeta *Imágenes*: Pestaña *General*



(b) Propiedades de la carpeta *Imágenes*: Pestaña *Compartir*



(c) Propiedades de la carpeta *Imágenes*: Opciones avanzadas de la pestaña *General*

Figura 2.8. Permisos de ficheros y directorios en Windows XP

miento y presentación de las aplicaciones (todas presentan el mismo aspecto y algunas opciones de menú y botones en las ventanas comunes a todas ellas).

¿Sabías que. . . ?

UNIX es un sistema multiusuario y multitarea que comenzó a desarrollarse a finales de los sesenta, tomó “forma” durante los setenta y alcanzó su madurez y pleno desarrollo durante la década de los ochenta.

continúa en la página siguiente . . .

. . . viene de la página anterior

UNIX ha sido considerado, tradicionalmente, el S.O. de los grandes ordenadores: máquinas de cálculo intensivo, *mainframes* o estaciones de trabajo habitualmente presentes en los centros de proceso de datos o en los centros de cálculo de grandes empresas, universidades y organismos internacionales.

En la actualidad también disponemos de “versiones” UNIX para PC (p.e. Linux y FreeBSD), algo ciertamente demandado, dados los grandes avances realizados en los últimos años en este tipo de arquitectura.

Sólo tienes que recordar una cosa: en Windows, para abrir una carpeta o ejecutar un programa, hacemos *doble clic* sobre el icono que la/lo representa. Pero en KDE esto no es así. Por defecto, para abrir una carpeta, acceder a un dispositivo o ejecutar un programa basta con hacer *un único clic* sobre el icono correspondiente.

Debes tener en cuenta igualmente que en la configuración de KDE existe una opción que le confiere casi idénticas características en su funcionamiento y un aspecto similar a Windows: por ejemplo, puedes usar doble clic para abrir un fichero o directorio en lugar de un único clic.

2.4.1. Directorios estándar

En Unix la jerarquía del sistema de ficheros tiene un nodo principal: el directorio raíz. El directorio raíz no tiene nombre y nos referimos a él con una barra (/).

Hitos en el desarrollo de Unix (hasta la aparición de Linux)

1965: Se diseña un S.O. multiusuario muy ambicioso: *MULTICS* (MULTiplexed Information and Computing Service). Participan, entre otros, MIT, General Electric y Bell Labs de AT&T. Se pretendía dar capacidad de cálculo siguiendo el modelo de las compañías eléctricas: gran número de usuarios se conectarían a un ordenador central (aproximadamente de la potencia de un PC-AT).

1968: El proyecto fracasa. Bell Labs se retira de MULTICS y uno de sus empleados, Ken Thompson (involucrado en el proyecto), empezó a escribir para un PDP-7, una versión muy simplificada de MULTICS, escrita en lenguaje ensamblador.

1969: Dennis Ritchie y Rudd Canaday se unen a Thompson en su proyecto. Esta versión funcionaba bien y uno de los colegas de Thompson la llamó *UNICS* (UNiplexed Information and Computing Service).

1970-1974: Brian Kernighan lo bautiza *UNIX*. Se reescribe en B, un lenguaje inventado por Thompson, pero con ciertas carencias. Ritchie diseñó, a partir de él, C. Él y Thompson reescribieron UNIX en C para PDP-11, que permitía tener varios usuarios simultáneamente, al proteger la memoria por hardware. Además se incluyen innovadoras herramientas de procesamiento de texto.

1974-1979: Bell Labs publica la descripción de UNIX: S.O. portable, escrito en lenguaje de alto nivel (C), multiusuario, multitarea, sistema de archivos jerárquico y con editor, compilador de C y herramientas de procesamiento de texto.

continúa en la página siguiente . . .

. . . viene de la página anterior

Las universidades manifiestan interés por compartirlo. Bell lo distribuye *con fuentes* de manera *abierta y libre* a instituciones académicas y científicas. Sólo obliga a reportar investigaciones y mejoras.

Al utilizarse en las universidades más importantes de Europa y EE.UU. y haber cada vez más gente trabajando en su desarrollo "progresa" muy rápidamente. Aparece la primera versión "estándar": la versión 6.

1979: Se licencia UNIX versión 7, la cual se populariza enormemente como plataforma para experimentar y desarrollar tecnología, probar algoritmos, protocolos de comunicación, lenguajes de programación y gestores de bases de datos. Por entonces ya estaba ampliamente distribuido por gran cantidad de estaciones de trabajo y muchos estudiantes se habían formado a partir de UNIX.

1980-1983: La universidad de Berkeley proporciona UNIX BSD 4.0 al departamento de defensa de EE.UU. (BSD es la versión de UNIX de Berkeley). En ARPANET comienza a utilizarse TCP/IP bajo UNIX BSD 4.0. UNIX System III constituye el primer intento de unificación de versiones.

1980-1983: Aparecen versiones comerciales: Xenix, UNIX para arquitectura IBM S/370. Otro estándar: System V y sus diversas *releases*. SUN, HP y Silicon Graphics, entre otros, lanzan estaciones de trabajo bajo UNIX dedicadas al procesamiento gráfico profesional y CAD.

1984: UNIX System V Release 2 incorpora los últimos avances tecnológicos en manipulación y administración de recursos, protocolos de comunicación, lenguajes, bases de datos, procesamiento distribuido y paralelismo. El proyecto de interfaz gráfica de usuario Athena X-Window System se desarrolla e implementa bajo UNIX.

1985-1986: Xenix System V para PC (80286 y 80386). SUN implementa NFS (Network File System) sobre TCP/IP para compartir volúmenes (discos) en red y hace pública la definición de los protocolos necesarios. La red de la fundación nacional de ciencia de EE.UU. y Arpanet utilizan TCP/IP a plenitud: nace *Internet*. Se publican críticas a UNIX para tratar de frenar su avance. TCP/IP se convierte en el protocolo estándar de comunicación en todo tipo de redes.

1987: Ante las diferencias entre las distintas versiones, la IEEE definió el estándar POSIX, que intenta ser una combinación entre System V Release 3 y BSD 4.3.

1988: *X Window System* se proclama sistema estándar para el desarrollo de interfaces y aplicaciones gráficas. Andrew Tanenbaum publica un libro sobre sistemas operativos acompañado de una versión de UNIX basada en POSIX, escrita por él y llamada *Minix*.

1989: UNIX e Internet caminan de la mano: TCP/IP, NFS, e-mail. . .

1991: Un estudiante universitario en Finlandia llamado Linus Torvalds está trabajando en modificaciones a Minix. Solicitando información por correo electrónico en Internet sobre la definición y contenido del estándar POSIX, recoge apoyo de curiosos e interesados en su proyecto: *Linux*, una versión de UNIX POSIX de libre distribución. Las consideraciones de diseño del núcleo del sistema Linux son portabilidad y eficiencia. Internet está gestando la WWW para todo el mundo (UNIX).

Los sistemas operativos de la familia UNIX siguen, normalmente, ciertas "conven-

ciones” en los nombres de algunas de las carpetas donde se localizan los ficheros de configuración del sistema, programas ejecutables, librerías, documentación, ficheros de configuración de aplicaciones, etc. Así:

/bin contiene los ejecutables básicos del sistema (órdenes para crear, copiar, mover. . . ficheros y carpetas, etc.).

/dev contiene los dispositivos susceptibles de ser conectados a nuestro ordenador, representados en forma de archivos especiales (en Unix, el ratón, cada disco duro, etc., llevan asociados sendos ficheros).

Si quieres saber más. . .

Una de las funciones principales de un S.O. consiste en controlar todos los dispositivos de entrada/salida de la computadora: debe emitir órdenes a los dispositivos, capturar interrupciones y manejar errores.

Pero también debe proporcionar una *interfaz* entre los dispositivos y el resto del sistema que sea simple y fácil de utilizar. Además, la interfaz debe ser la misma para todos los dispositivos, procurando conseguir la *independencia* de los mismos.

En general, los sistemas operativos de la familia UNIX “unifican” (en cierta manera) la gestión de la entrada y salida de datos con la del sistema de ficheros. Así, los dispositivos se representan mediante ficheros *especiales* en `/dev`.

Esto permite disponer de una interfaz única, independiente del dispositivo y una *facilidad* para los programadores: acceder a los dispositivos requiere las mismas llamadas al sistema que se usan para acceder, en general, a ficheros, ya que no hay que tener mecanismos especiales de escritura o lectura que dependan del dispositivo, todo se trata como si fuese un fichero.

/etc contiene ficheros para la configuración de aplicaciones, programas del sistema y servicios de la máquina (p.e. Internet) entre otras cosas. Las aplicaciones o servicios que requieran más de un fichero para su configuración, crean un subdirectorio para agruparlos.

/home directorio en el que se disponen los directorios de cada usuario del ordenador.

/lib lugar en el que se almacenan ficheros auxiliares del ordenador, necesarios para el funcionamiento de la mayoría de los programas instalados en el sistema (las denominadas librerías o bibliotecas).

/media directorio en el cual se “montan” los dispositivos *removibles* (CD, DVD, memorias y/o discos USB, iPods, etc.), a cuyo sistema de ficheros se quiere acceder. El sistema suele crear automáticamente una carpeta por cada (tipo de) dispositivo.

/mnt carpeta desde la cual se “montan” temporalmente sistemas de archivos de otros discos o sistemas remotos. Si se quiere poder montar varios volúmenes, podemos crear varias carpetas al efecto (lo detallamos más adelante).

/tmp directorio "temporal" del sistema. Es una carpeta que pueden usar los programas para crear ficheros dinámicamente (operaciones intermedias, etc.); esto es, ficheros auxiliares que pueden eliminarse sin peligro cuando apagamos el ordenador.

/usr contiene todo lo necesario para ejecutar programas de aplicación (o de usuario). Documentación, ejecutables, librerías, etc.

/var contiene ficheros de registro (bitácora), información del sistema, el correo de los usuarios, las "colas" de las impresoras, etc.

2.4.2. Rutas

Mediante el uso de carpetas (o directorios), la información se organiza en forma de *árbol*, de ahí la denominación "organización jerárquica". En este tipo de sistemas de ficheros:

- un nodo interno es un directorio,
- un nodo externo u hoja del árbol es un directorio *vacío* o un fichero,
- una rama es un apunte (anotación) en el directorio que hace referencia a:
  - otro directorio (subdirectorio), o
  - a un fichero;
- la raíz del árbol es la carpeta principal o directorio *raíz*.

Si llamamos *nombre simple* al nombre que tiene un fichero dentro del directorio que lo contiene, entonces llamaremos:

- *nombre absoluto* (ruta) de un fichero, a su nombre simple precedido de la secuencia de directorios que hay que abrir desde la raíz para encontrarlo.
- *nombre relativo* a un directorio cualquiera X de un fichero, a su nombre simple precedido de la secuencia de directorios que llevan al fichero *desde* X.

Dentro del sistema de ficheros, cada fichero debe ser referenciado (nombrado) de forma *no ambigua*. Esto quiere decir que no podrá haber dos ficheros con el *mismo nombre absoluto* en el sistema de ficheros.

También dentro del sistema de ficheros, el usuario siempre está trabajando en un directorio, al cual se le llama carpeta o directorio actual. El nombre relativo de un fichero siempre se interpreta como relativo al directorio actual. La Figura 2.9 muestra un ejemplo de un sistema de ficheros en Windows.

En la jerarquía ilustrada en la Figura 2.9,

Mi PC\C:\Asignaturas\509

es un nombre absoluto de carpeta. Un nombre absoluto de fichero sería, por ejemplo:

Mi PC\C:\Asignaturas\InfBas\intro.pdf

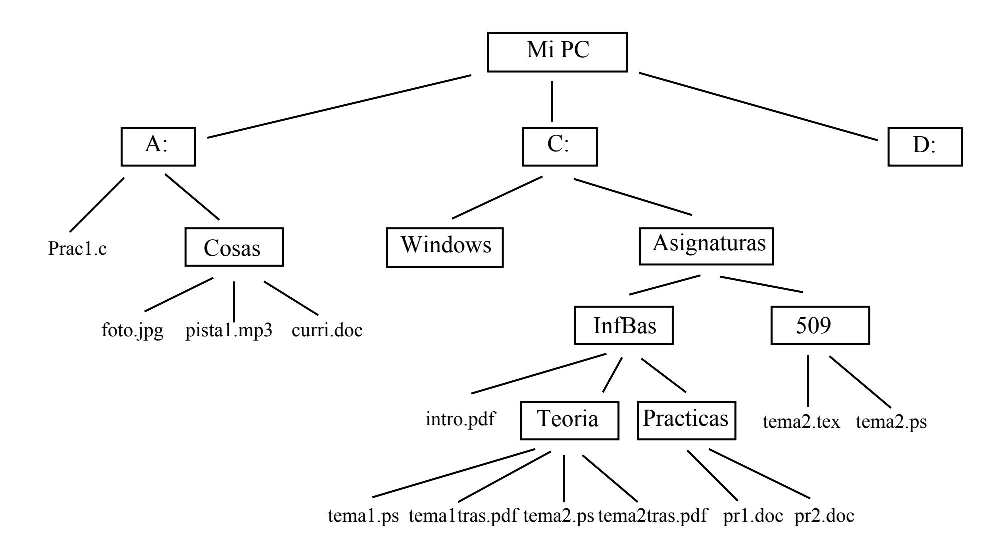


Figura 2.9. Ejemplo de sistema de ficheros en Windows

.

InfBas\Teoria es un nombre de carpeta relativo a la carpeta Asignaturas, e

InfBas\Practicas\pr1.doc

es un nombre de fichero igualmente relativo a la carpeta Asignaturas.

Si el directorio actual de trabajo es Mi PC\C:\Asignaturas\InfBas, entonces

Teoria\tema2.ps y Practicas\pr2.doc

son nombres relativos al directorio actual, pero 509\tema2.tex no lo es. Existen nombres simples iguales en el ejemplo (tema2.ps), pero pueden existir así en el sistema porque tienen distinto nombre absoluto.

En el ejemplo anterior, el directorio raíz es Mi PC, y así se ha considerado en los ejemplos de nombres absolutos, situándolo como primer directorio de todos los caminos. Sin embargo, en la práctica Mi PC no aparece como primer directorio de ninguno de los caminos, siendo el directorio raíz, en realidad, el *nombre de la unidad* de almacenamiento secundario (A:, C:, D:...).

La razón de ello es que cada unidad de almacenamiento en Windows define su *propio* sistema jerárquico de ficheros y el nombre de la unidad es la raíz real de cada sistema. Mi PC es una raíz virtual que se presenta en el escritorio para aparentar un único sistema de ficheros global. Esto es, el sistema de ficheros *no* es único en Windows.

Por consiguiente, en Windows los nombres absolutos comenzarán desde el nombre de la unidad de almacenamiento secundario en el que se encuentren los respectivos elementos (carpetas o ficheros) que referencian.

En el caso de Unix, el concepto de ruta es el mismo que para Windows. Simplemente cambia el separador de carpetas; ya no es \, sino /. Los sistemas de directorios de UNIX y de MSDOS/Windows son similares: en ambos se permiten rutas absolutas y relativas y se tienen las entradas . y .. en cada directorio para representar, respectivamente, al directorio actual de trabajo y al directorio que lo contiene (el "padre"). Sin

embargo, hay importantes diferencias entre los dos sistemas de ficheros. Como hemos visto, en Unix las rutas absolutas comienzan con / (directorio raíz; el primero que existe en la jerarquía, ya que en Unix el sistema de ficheros es único) mientras que en Windows comienzan con la letra que identifica la unidad, seguida de *dos puntos* (:) y \ (C:\, por ejemplo).

Cuando te conectas a un ordenador Unix has de identificarte proporcionando un nombre de usuario (`login`) y una contraseña secreta (`password`). Si se introducen ambos correctamente, se inicia una sesión de trabajo. Si te conectas a una consola en modo texto, el sistema pedirá una orden mostrando un *prompt*, que aquí representaremos con un dólar (`$`). En el caso de trabajar en un entorno gráfico (como KDE), aparecerá el escritorio correspondiente con sus menús y demás.

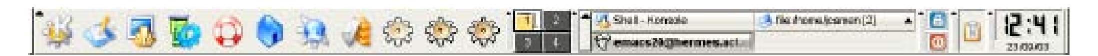


Figura 2.10. La barra de tareas del escritorio KDE 3.1.3 (Debian)

Cada usuario tiene un directorio personal (normalmente) en `/home/` cuyo nombre es igual al nombre del usuario. Supongamos que tu nombre de usuario es `juan`, tu directorio personal es `/home/juan`. Al iniciar una sesión de trabajo "entras" directamente a tu directorio personal. Decimos que tu directorio personal es, en ese momento, tu *directorio activo*.

El directorio activo permite evitar nombres de ruta largos al referirse a un fichero: nos podemos referir al fichero `/home/juan/datos.txt` por su nombre, `datos.txt`, si el directorio activo es `/home/juan/`.

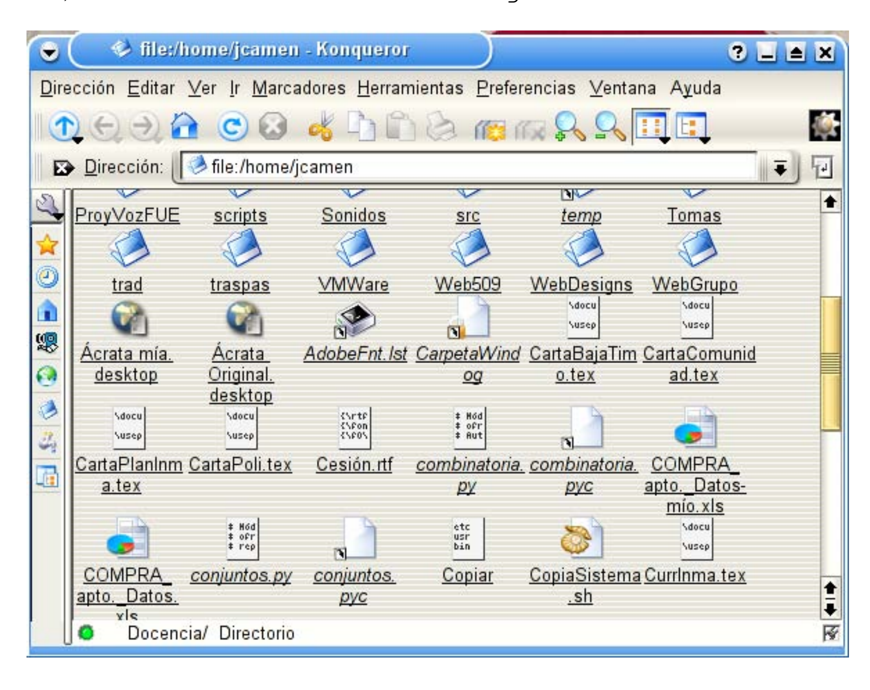


Figura 2.11. Konqueror mostrando el contenido de un directorio personal

Recuerda que si una ruta no empieza por / decimos que es una *ruta relativa*. En caso contrario es una *ruta absoluta*. Una ruta relativa añade implícitamente la ruta del directorio activo para formar así la ruta completa de un fichero.

En KDE existe un navegador por defecto que, además, está integrado con el sistema de ficheros y permite "navegar" por sus contenidos (como Internet Explorer en Windows): *Konqueror*.

En la barra de tareas (véase la Figura 2.10) encontrarás un icono que te permitirá arrancarlo, mostrando los contenidos de tu directorio personal. Dicho icono tiene habitualmente forma de "casita".

2.4.3. Cambio de directorio

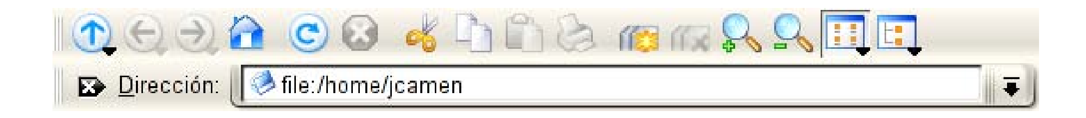


Figura 2.12. La barra de navegación (arriba) y la barra de dirección de Konqueror

Cambiar de directorio es tan sencillo como hacer clic con el botón izquierdo del ratón sobre la carpeta que lo representa. El directorio activo es el que muestra la ventana de Konqueror (`file:/home/jcamen/Web509`); véase la Figura 2.13. La barra de navegación de Konqueror (véase la Figura 2.12) permite ir atrás y adelante (navegar) en la jerarquía de directorios. Debajo de ella (véase la misma Figura 2.12) está la barra de dirección, que muestra el directorio cuyo contenido se está mostrando.

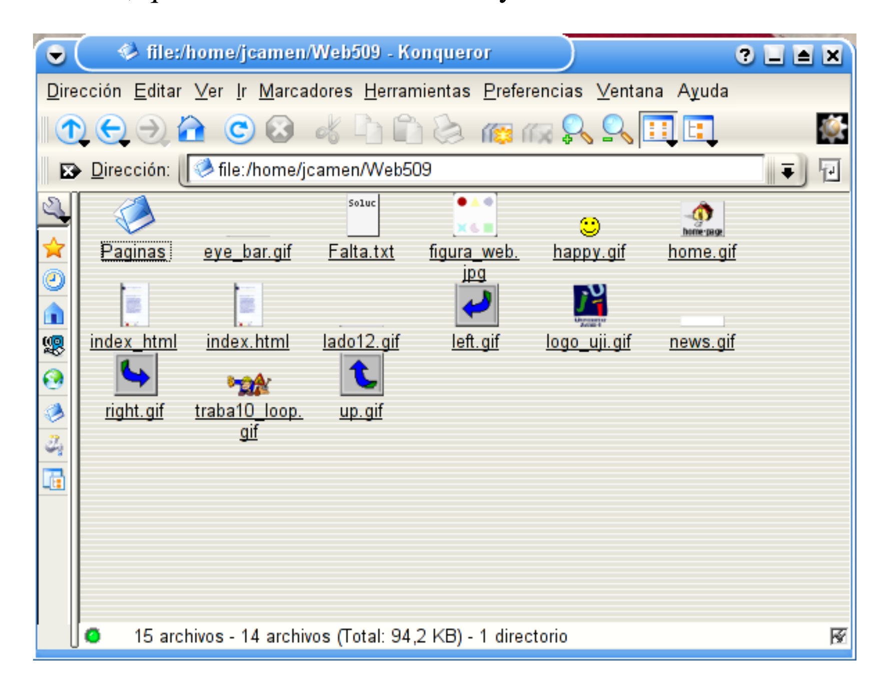


Figura 2.13. Konqueror mostrando el contenido de la carpeta Web509

Konqueror es un navegador, lo cual quiere decir que si escribimos en esta barra la dirección de una página web (como `http://www.uji.es`) entonces se mostrará el contenido de dicha página[9] y los botones de la barra de navegación se comportarán de la forma "habitual" (adelante, atrás, recargar página, ir a la página de inicio, etc.).

[9]Suponiendo que tengamos conexión a Internet, claro está.

Haciendo clic en el icono de la barra de navegación con forma de casa, Konqueror mostrará el contenido del directorio inicial del usuario ya que así es como está configurado por defecto (véase la Figura 2.11).

2.4.4. Listar el contenido de un directorio

Cada vez que accedemos a una carpeta, Konqueror muestra sus contenidos. Por defecto, Konqueror lista primero los subdirectorios (iconos con forma de carpeta) y, a continuación, los ficheros. Normalmente se emplea el orden alfabético, sin distinguir entre mayúsculas y minúsculas (véase la Figura 2.11).

2.4.5. Creación de ficheros

Hay muchos modos de crear ficheros. Lo más normal es que los crees como resultado de usar un programa:

- un editor de textos permite crear y modificar ficheros con texto,
- un editor gráfico permite crear y modificar ficheros con imágenes,
- un conversor MP3 permite crear un fichero de música comprimida (MP3) a partir de un fichero de música sin comprimir (por ejemplo, en formato WAV),
- etc.

2.4.6. Creación de directorios

Crear un directorio es una tarea muy sencilla con Konqueror (más o menos como con Internet Explorer en Windows). Basta con elegir una zona "despejada" de la ventana de Konqueror que muestre el contenido del directorio donde quiero crearlo y hacer clic con el botón derecho del ratón. Surge un menú contextual cuya primera opción se titula *Crear nuevo*. Dicha opción despliega un submenú cuya primera opción es la que permite crear directorios. También podemos llegar a esta opción a través de la opción *Crear nuevo* del menú *Editar*.

Figura 2.14. Menú contextual de Konqueror

2.4.7. Usuario y grupo propietarios

Cada fichero y directorio del sistema pertenecen a un *usuario* y *grupo* determinados: el usuario y el grupo propietarios. Éstos, junto con los permisos, determinan:

- *quién puede leer* el contenido de un fichero o directorio,
- *quién puede modificar* el contenido de un fichero o directorio,
- *quién puede ejecutar* un fichero de programa o acceder al contenido de un directorio.

Se puede cambiar el propietario y grupo de un fichero si se tiene permiso para ello.

2.4.8. Permisos

Acabamos de aprender que todos los ficheros y directorios en Unix tienen un propietario y un grupo. Asociados a éstos tienen unos permisos, que determinan qué se puede hacer. Los ficheros tienen tres tipos de permiso diferentes, que se representan con una letra[10]:

- `r`: permiso de *lectura* (por *read*). ¿Se puede ver el contenido del fichero?
- `w`: permiso de *escritura* (por *write*). ¿Se puede modificar el contenido del fichero? (y eso implica poder borrarlo).
- `x`: permiso de *ejecución* (por *execute*). ¿Se puede ejecutar el fichero? (sólo tiene sentido si el fichero es un programa ejecutable).

Los permisos se representan mediante 9 bits, agrupados en tres grupos de tres bits de la siguiente forma:

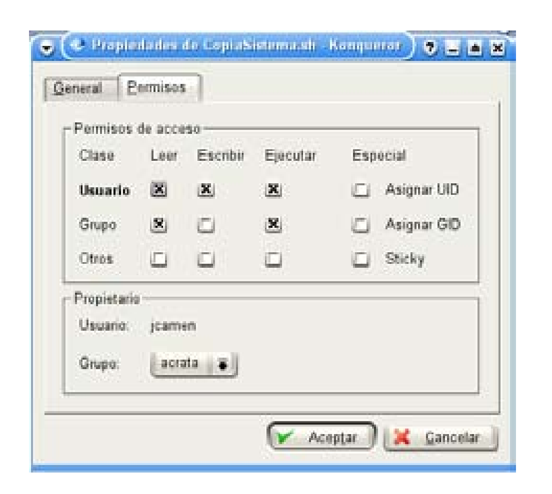
- Cada grupo fija los permisos de lectura (`r`), escritura (`w`) y ejecución (`x`).
- El primer grupo de 3 bits los fija para el *usuario* propietario del fichero: `u` (por *user*).
- El segundo grupo de 3 bits, para todos los usuarios adscritos al *grupo* propietario del fichero: `g` (por *group*).
- El tercer grupo de 3 bits, para el resto de usuarios (*otros*) registrados en el sistema: `o` (por *others*).

Cuando se trata de un *directorio*, los permisos se interpretan con ciertos matices:

- `r`: permiso de *listado*. ¿Se puede listar el contenido del directorio?
- `w`: permiso de *modificación*. ¿Se puede añadir (creándolo, por ejemplo) o eliminar un fichero del directorio?
- `x`: permiso de *lectura de ficheros*. ¿Se puede leer el contenido de un *fichero del directorio* si, además, tenemos permiso de lectura sobre él? (es decir, este permiso nos da acceso a los contenidos del directorio; pero recuerda que los contenidos llevan sus propios permisos).

---

[10]Esto tiene más sentido cuando trabajamos con un intérprete de órdenes. En el capítulo 6 verás para qué sirven estas letras.



(a) Konqueror muestra los permisos del fichero *CopiaSistema.sh*



(b) Konqueror muestra los permisos del directorio *figuras*

Figura 2.15. Permisos en ficheros y directorios

Podemos cambiar los permisos de un fichero o directorio usando Konqueror. Para ello, en primer lugar hay que seleccionar el fichero o directorio cuyos permisos deseamos modificar. Después, ejecutamos la opción *Propiedades* del menú *Editar* (también podemos acceder a esta opción a través del menú contextual del objeto seleccionado[11]). Finalmente, seleccionamos la pestaña *Permisos* de la ventana que aparece y hacemos clic en el botón *Permisos avanzados*. Veamos un par de ejemplos:



(a) Quitamos permisos de ejecución del fichero *CopiaSistema.sh* a los miembros del grupo



(b) Damos permisos de escritura a los miembros del grupo y de lectura a otros en el fichero *CopiaSistema.sh*

Figura 2.16. Cambiando permisos con Konqueror: un par de ejemplos

[11]Este menú es el que se despliega al hacer clic con el *botón derecho* del ratón sobre el objeto.

2.4.9. Más sobre rutas

Es tan frecuente referirse al directorio personal propio que disponemos de una abreviatura para él: ˜. Para el ejemplo de la figura siguiente, si pulsamos *Intro*, Konqueror mostrará el contenido del directorio personal.

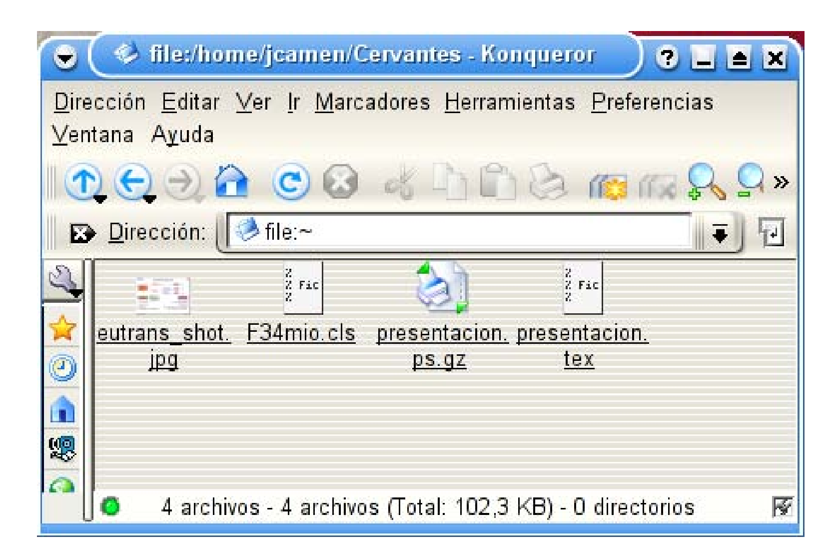


Figura 2.17. El símbolo $\sim$ representa el directorio personal del usuario

2.4.10. Copia y cambio de ubicación de ficheros y directorios

Konqueror también puede utilizarse para copiar y/o cambiar de ubicación ficheros y directorios enteros con todo su contenido[12]. Para ello basta, en primer lugar, con seleccionar el (los) fichero(s) o directorio(s) a copiar (mover). A continuación, abrimos otra ventana de Konqueror que muestre el contenido del directorio a dónde queremos copiar (mover) la selección efectuada. Finalmente, arrastramos la selección desde la ventana "origen" a la ventana "destino". En ese momento aparece un pequeño diálogo que muestra cuatro opciones. A nosotros nos interesan las opciones *Copiar aquí* (para copiar), *Mover aquí* (para el cambio de ubicación; esto es, copiar en el directorio destino y borrar del directorio origen) y *Cancelar* (abortamos la operación). La Figura 2.18 ilustra el proceso de copiar los ficheros `CartaBajaTimo.tex` y `CopiaSistema.sh` y el directorio `VMWare` y todo su contenido desde `/home/jcamen` hasta `/home/jcamen/Docencia/IG01/Curso03-04`. La operación de mover se efectuaría de forma análoga (eligiendo la opción correspondiente en el diálogo).

Para seleccionar podemos combinar la utilización del ratón con el empleo de la tecla *Shift* (mayúscula) para elegir ficheros adyacentes o bien con el empleo de la tecla *Ctrl* para añadir ficheros sueltos a la selección.

Puesto que el nombre de un fichero o directorio forma parte de su *ruta*, cambiar, simplemente, dicho nombre supone un caso particular de "cambio de ubicación". Para ello, en Konqueror existe la opción *Renombrar*. A ella podemos acceder a través del

[12]Igualmente, de manera similar a como lo permite Internet Explorer en Windows.



Figura 2.18. Copiando ficheros y directorios mediante Konqueror

menú *Editar* o bien a través del menú contextual (haciendo clic con el botón derecho del ratón) del fichero o directorio cuyo nombre deseamos cambiar. En ambos casos, tenemos que seleccionar el fichero o directorio previamente. Al ejecutar la opción *Renombrar*, observaremos que un recuadro rodea el nombre del fichero o directorio y que el cursor comienza a parpadear al comienzo del mismo indicando que podemos proceder al cambio de nombre.

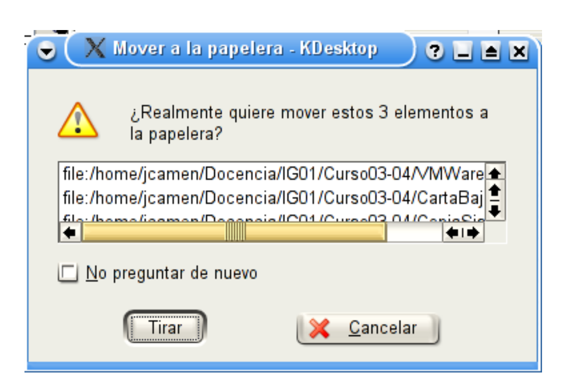


Figura 2.19. Confirmar la acción de tirar objetos a la papelera

Para finalizar esta sección, queremos apuntar la posibilidad de usar las opciones *Cortar*, *Copiar* y *Pegar* del menú *Editar* de Konqueror. *Cortar* significa trasladar (borrar) los objetos seleccionados del directorio donde están situados a una zona “especial” de KDE. *Copiar* significa hacer una copia de los objetos seleccionados a esa zona “especial” de KDE. Finalmente, *Pegar* realiza la operación de trasladar los objetos de esa zona “especial” de KDE al directorio cuyos contenidos muestra la ventana de Konqueror en la cual hemos ejecutado esta opción.

Así, copiar (mover) los ficheros y/o directorios seleccionados de, por ejemplo, `/home/jcamen` a `/home/jcamen/Docencia/IG01/Curso03-04` (otro directorio) se puede hacer ejecutando la opción *Copiar* (*Cortar*) del menú *Editar* de la ventana de Konqueror que muestra el contenido de `/home/jcamen` y, posterior-

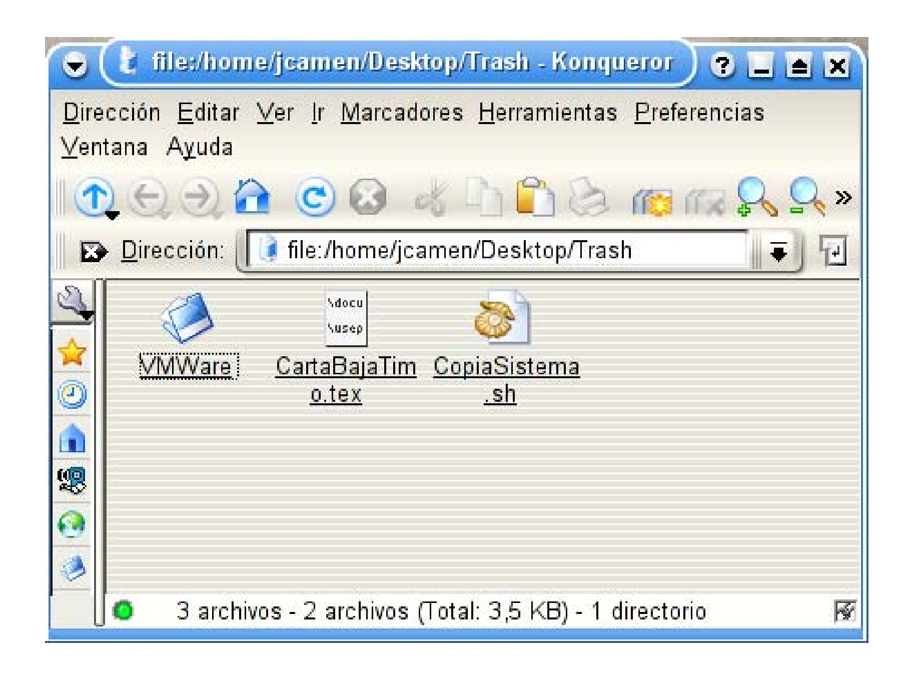


Figura 2.20. Konqueror muestra el contenido de la papelera

mente, ejecutando la opción *Pegar* del menú *Editar* en la ventana de Konqueror que muestra el contenido de `/home/jcamen/Docencia/IG01/Curso03-04`.

2.4.11. Eliminación de ficheros y directorios

También podemos emplear Konqueror para borrar ficheros y/o directorios. Podrás apreciar que, dentro del escritorio KDE, existe un icono que representa una *papelera* (por defecto, vacía). De manera intuitiva[13] podemos seleccionar aquellos objetos que deseemos eliminar, arrastrarlos y soltarlos encima de la papelera. En ese momento aparecerá el diálogo que muestra la Figura 2.19. Ahí podemos elegir entre realizar la acción de tirar a la papelera o bien cancelar la operación. Observa que existe un cuadrito que, si lo marcas, provocará que *no vuelva a aparecer* más este diálogo.

Cabe resaltar que tirar un objeto a la papelera *no implica necesariamente su eliminación*. Cualquier fichero y/o directorio que esté en la papelera puede ser recuperado. Para ello, basta con hacer clic en la papelera (abriendo una ventana de Konqueror que muestra su contenido) y *mover* (véase la sección anterior) el (los) fichero(s) y/o directorio(s) que se desea(n) recuperar al directorio correspondiente.

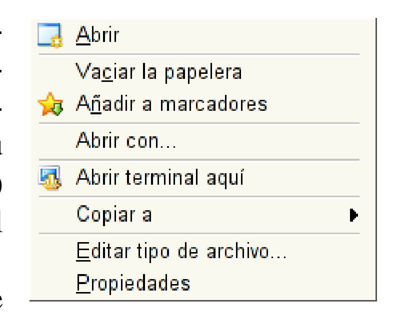


Figura 2.21. Eliminando el contenido de la papelera

De hecho, la papelera de KDE no es más que un directorio llamado `Trash` dentro del directorio `Desktop` que se encuentra en el directorio personal del usuario (véase la Figura 2.20). Debes tener en cuenta, no obstante, que en versiones de KDE posteriores a la 3.3 (de unos años hacia acá), el

[13] Tal como se hace, por ejemplo, en Windows.

directorio `Trash`, junto con cierta información "especial" acerca de los ficheros que contiene (ruta donde se encontraban originalmente, entre otros), se encuentra en la ruta `.local/share/` a partir del directorio personal del usuario. El KDE que usarás en las sesiones de prácticas es relativamente reciente, por lo que encontrarás el directorio `Trash` dentro de `.local/share/` normalmente.

Para eliminar definitivamente los objetos almacenados en la papelera basta con ejecutar la opción *Vaciar la papelera* del menú contextual que aparece al hacer clic *con el botón derecho* del ratón sobre el icono que representa la misma, tal como indica la Figura 2.21.

También podemos echar a la papelera los objetos seleccionados ejecutando la opción *Mover a la papelera* del menú *Editar*. Si lo que queremos realmente es *eliminar definitivamente* los objetos seleccionados (sin pasar por la papelera), entonces bastaría con ejecutar la opción *Eliminar* del menú *Editar*.

2.4.12. Edición de ficheros

La edición del contenido de un fichero se realiza a través de aplicaciones específicas para el tipo de fichero que deseamos crear o modificar (imágenes, sonido, texto, bases de datos, películas...). KDE asocia, por defecto, una (o varias) aplicación(es) que permite(n) "visualizar" (reproducir) y/o editar los ficheros según su tipo. Así, por ejemplo, asocia un programa multimedia (como `xmms`) a ficheros MP3, un reproductor de toda clase de películas (como `mplayer`) a ficheros AVI, un editor de textos (como `emacs`) a los ficheros de texto, un programa de hoja de cálculo (como `Calc` de OpenOffice) a ficheros que contienen hojas de cálculo, etc. Cualquier usuario puede cambiar las asociaciones establecidas por KDE en su escritorio. Para ello basta con seleccionar un fichero del tipo que deseamos cambiar (por ejemplo, PDF). Después habría que ejecutar la opción *Editar tipo de archivo* del menú *Editar* de la ventana de Konqueror. La Figura 2.22 muestra la ventana que aparecería.

El tipo de fichero fundamental que usaremos en el curso (en ésta y otras asignaturas) es el *fichero de texto* (por ejemplo, el código fuente –las instrucciones– de los programas se almacena en ficheros de texto). El contenido de un fichero de texto se crea y modifica con un *editor de textos*. Hay infinidad de editores de textos. Varían en facilidad de uso y potencia. En el entorno Unix hay dos editores muy populares:

- `vi`: el "editor visual". Es un editor muy antiguo (data de los años setenta). Su manejo es bastante complejo. Se sigue usando porque todas las versiones de Unix lo ofrecen (ocupa muy poco espacio en disco y consume pocos recursos –memoria).

- `emacs`: es un editor algo más moderno (años ochenta). Su manejo es algo más fácil que el de `vi`. Su característica más importante es la capacidad de extender su funcionalidad para adaptarlo a tareas muy concretas. En general, consume más recursos (memoria y tiempo de procesador) que `vi`.

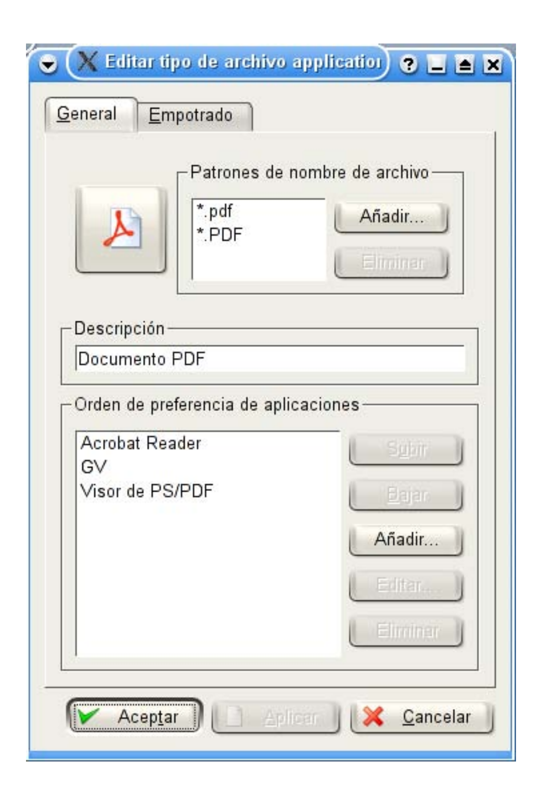


Figura 2.22. Ventana para modificar la asociación establecida por KDE entre aplicaciones y tipos de fichero

Podrás ejecutar `emacs` (o su variante `xemacs`) a través del menú que se despliega al hacer clic con el ratón sobre el botón K de la barra de tareas de KDE. Lo encontrarás normalmente en el submenú *Editores* o similar. Después abres el fichero cuyo contenido deseas editar mediante la opción *Open file...* del menú *Files* de esta aplicación.

Muchos usuarios de Linux encuentran más cómodo usar una consola en modo texto (intérprete de órdenes) para ejecutar `emacs` (`xemacs`). En el capítulo 6 veremos más cosas respecto al intérprete de órdenes de Unix. De momento, deberías saber que puedes arrancar una consola en modo texto haciendo clic con el ratón sobre el icono que presenta una pantalla de ordenador con una concha encima en la barra de tareas. También puedes ejecutar la opción *Konsole (Programa de terminal)* del submenú *Sistema* que aparece al desplegar el menú del botón K.

Entonces, para editar un fichero basta con teclear la orden `vi` o `emacs` (`xemacs`) seguido del nombre del fichero en dicha consola. Si no existe, el editor lo crea:

```
$ vi novela.txt
$ emacs redaccion.txt
```

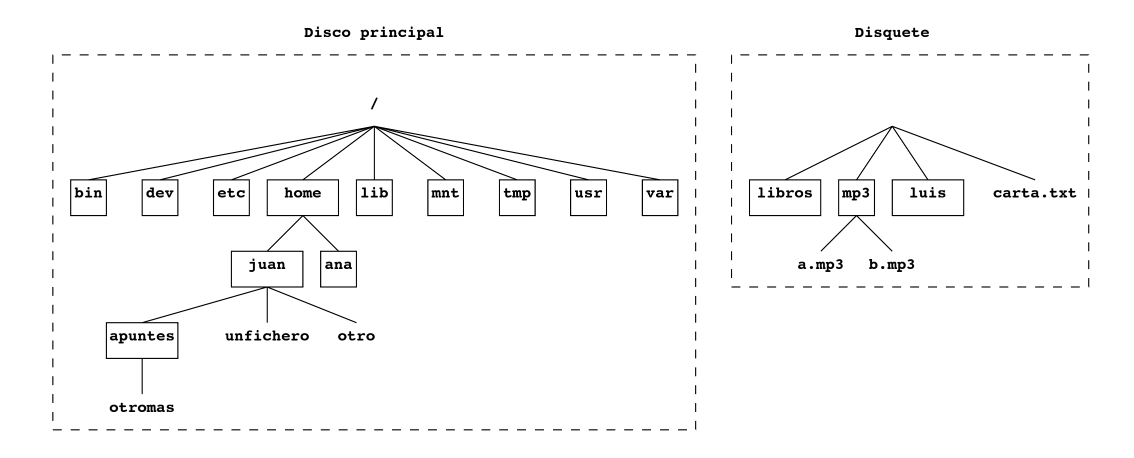


Figura 2.23. Jerarquía de ficheros en un disco duro y en un disquete

Aprenderás a utilizar el editor `emacs` en clase de prácticas (en realidad, una versión de `emacs` llamada `xemacs`). Por otro lado, observa que puedes, meramente, consultar el contenido de un fichero de texto abriéndolo con el editor correspondiente.

2.5. UNIDADES

Las unidades (discos duros, disquetes) suelen tener, cada una, su propia jerarquía de directorios y ficheros; es decir, tienen un sistema de ficheros completo (véase la Figura 2.23). Windows ofrece cada sistema de ficheros (jerarquía) por separado. Se accede a cada una de las unidades[14] con letras:

- `A:` suele ser el disquete.
- `B:` está reservado, por cuestiones "históricas"[15], a una segunda unidad de disquete.
- `C:` es el disco duro principal.
- `D:` suele ser otro disco duro o un CD-ROM (o DVD, CD-RW, etc).
- `E:` suele ser otro CD-ROM (o DVD, CD-RW, etc).

Figura 2.24. Unidades en KDE

No obstante, los sistemas operativos de la familia UNIX *sí* ofrecen una jerarquía de ficheros única. Luego, ¿qué hacemos cuando queremos acceder a otras unidades distintas del disco principal?

Es posible vincular una unidad a un directorio Unix. A esa acción se la denomina *montaje*. Por tanto, en un determinado directorio, se puede "colgar" la jerarquía de ficheros y directorios de una unidad. A partir de ese momento, los ficheros y directorios de la unidad montada son indistinguibles (salvo cosas como la velocidad) de los del disco principal.

El escritorio KDE muestra iconos que representan las unidades que existen en nuestro PC. En el ejemplo mostrado en la Figura 2.24 se puede apreciar un icono con forma de disquete (representa la disquetera), dos con forma de disco compacto (representan una unidad convencional de CD-ROM –EIDE– y otra SCSI[16]) y, finalmente, otro con forma de caja gris[17] –disco duro– (representa la partición Windows).

Para acceder a ("montar") una unidad en KDE[18] (por ejemplo, la disquetera) basta con hacer clic con el ratón sobre el icono que la representa. En ese instante aparecerá una ventana de Konqueror mostrando sus contenidos. Puedes observar que la unidad está montada porque KDE añade una lucecita verde al lado del icono que la

[14]Mejor dicho, al sistema de ficheros de cada unidad.

[15]Que tienen que ver con el hecho de que las primeras generaciones de PC no tenían disco duro. Sin embargo, muchos PC venían equipados con dos disqueteras.

[16]Véase el capítulo 4 que versa sobre hardware.

[17]El icono de arriba representa la papelera, que no es una unidad.

[18]En el caso de, por ejemplo, disquetera y CD-ROM debemos introducir previamente el disquete o CD, como resulta obvio.

representa. A partir de ese momento puedes trabajar con esa unidad como si se tratase de un directorio más de la jerarquía de Linux. Típicamente las unidades se montan en subdirectorios de `/media` (lo más normal, últimamente) o `/mnt`.

- `/media/floppy` suele ser el disquete.
- `/mnt/windows` suele ser el disco en el que tenemos instalado Windows (si tenemos Windows y Linux en la misma máquina).
- `/media/cdrom` suele ser el CD-ROM.
- `/media/dvd` suele ser el DVD.

Cuando hayas terminado de trabajar con la unidad montada *debes tener la precaución de desmontarla*. Y si se trata de un dispositivo *removible* (como un disquete o CD)[19] tienes que desmontar *antes de sacar el dispositivo*. Si no procedes así corres el riesgo de perder datos. Para desmontar basta con hacer clic con el botón derecho del ratón sobre el icono que representa la unidad montada. Observarás que, en el menú contextual que aparece, hay una opción denominada *Desmontar*. Ejecútala y ya está, aunque no olvides cerrar previamente la ventana Konqueror que muestra los contenidos de esa unidad. En el caso de un CD-ROM verás que aparece, además, la opción *Expulsar*, dentro del menú *Acciones*. Esta opción no sólo desmonta el dispositivo, sino que también expulsa el CD fuera de la unidad.

Sin embargo, algunos usuarios prefieren usar el intérprete de órdenes, mediante una consola en modo texto (véase la sección 2.4.12), para montar y desmontar unidades. Precisamente, para montar disponemos de la orden `mount`. Por ejemplo, para montar un disquete y así poder tener acceso a su sistema de ficheros, has de ejecutar:

```
$ mount /mnt/floppy , o bien, mount /media/floppy
```

Podemos apreciar el resultado final en la Figura 2.25, teniendo en cuenta la situación mostrada anteriormente en la Figura 2.23.

¿Sabías que. . . ?

En Unix existen, *grosso modo*, dos tipos de dispositivos:

De bloques, que están divididos en una serie de bloques numerados, permitiendo acceder a ellos de forma individual. El ejemplo más claro son los discos duros.

De caracteres, que conceptualmente se ven como una corriente de bytes que se leen o escriben secuencialmente. Ejemplos de estos dispositivos son el ratón o el teclado.

[19] Realmente no puedes sacar un CD si previamente no lo desmontas, ya que Linux bloquea la unidad de CD-ROM y no deja abrirla hasta que el CD ha sido desmontado.

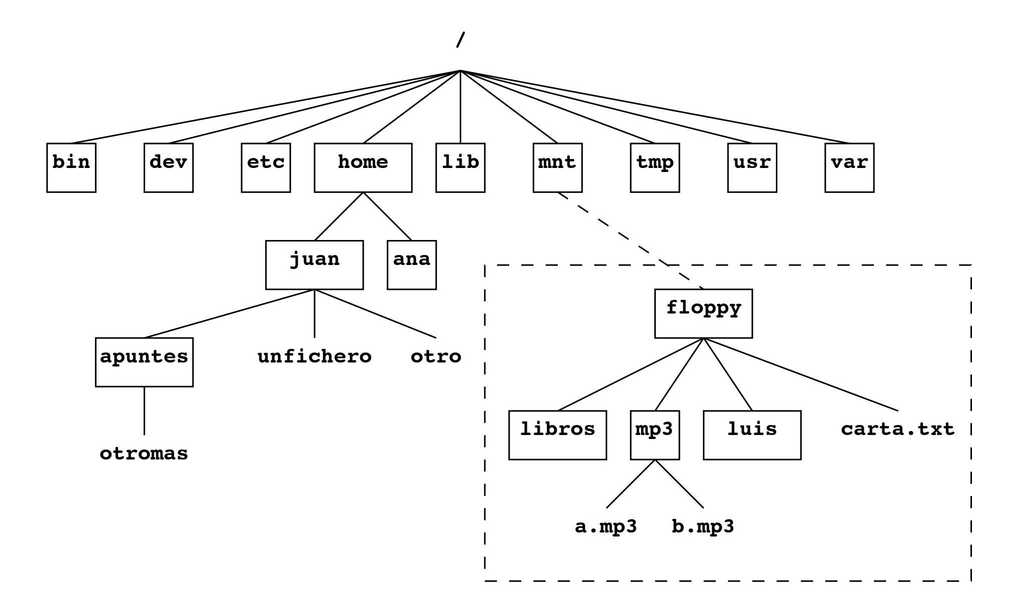


Figura 2.25. Contenidos del disquete "montados" en el sistema de ficheros principal de un sistema Linux

Aunque insertes el disquete en la disquetera, si no lo montas, Unix no lo "ve". Recuerda que cuando hayas acabado con el disquete, has de *desmontarlo* antes de retirarlo de la disquetera. Si no lo haces, puedes perder datos. La orden para desmontar es `umount` (de *unmount*).

```
$ umount /mnt/floppy , o bien, umount /media/floppy
```

2.6. FORMATOS DE SISTEMAS DE FICHEROS

Ya sabemos que los sistemas de ficheros son una abstracción que nos permite organizar los datos en un disco duro. Pero, por ejemplo, un disco duro es una superficie magnética que memoriza unos y ceros. Hay varias formas de escribir/leer los unos y ceros para que se interpreten correctamente como ficheros y directorios. Decimos que hay varios *formatos* o *tipos* de sistemas de ficheros.

No existe un tipo único de formato para representar un sistema de ficheros. Esta es una cuestión que depende del S.O. que queramos utilizar (y a veces de los dispositivos). Diferentes sistemas operativos usan mecanismos internos diferentes y, en consecuencia, mecanismos distintos para representar los sistemas de ficheros con los que operan. Los tipos más comunes que podemos encontrarnos son:

FAT-msdos: usado por MS-DOS. Obsoleto. Es la base del VFAT o FAT extendido, usado en algunos sistemas Windows (95, 98, Me). Bastante limitado (por ejemplo, los nombres de ficheros y carpetas se limitan a 8 caracteres, más 3 para la extensión).

VFAT-FAT32: también conocido como FAT extendido. Usado por muchas unidades Windows. Básicamente es el FAT con soporte para nombres largos. Los disquetes suelen presentar ese formato.

ext2: el sistema de ficheros ext2 era el sistema de ficheros estándar de Linux. Ahora está obsoleto. Hoy en día ha sido sustituido por **ext3**, el cual es más avanzado y supone la siguiente generación en el sistema de ficheros tradicional de Linux[20]. Ambos soportan nombres largos.

ReiserFS: también propio de los sistemas Linux, aunque más moderno y avanzado que ext2.

hfs: éste es el sistema de ficheros estándar de los clásicos Apple Macintosh. Soporta nombres largos y ciertos caracteres especiales.

iso9660: es el sistema de ficheros utilizado en los CD-ROM, esto es, discos compactos de *datos*. Es muy limitado, pero afortunadamente existen "extensiones" como Rock Ridge (en Linux) y Joliet (en Windows) que permiten dotarlo de soporte para nombres largos, uso de mayúsculas y minúsculas, permisos, etc.

NTFS: sistema de ficheros "nacido" con Windows NT (NTFS significa New Technology File System). Otros Windows posteriores como Windows 2003 y Windows XP lo han adoptado. En estos sistemas se sugiere emplearlo[21] para mejorar el rendimiento. Es mucho más avanzado, potente y moderno que cualquiera de los sistemas basados en FAT.

Hay algunos más: como *udf*, el nuevo sistema de ficheros usado, fundamentalmente, en los DVD (Digital Video/Versatile Disc).

Completando información:

Windows soporta nativamente los sistemas FAT y VFAT. También soporta iso9660 con las extensiones denominadas *Joliet*.

Estas extensiones fueron desarrolladas por Microsoft y permiten la utilización de nombres largos en formato *unicode* (un código de caracteres de 16 bits que soporta casi todos los idiomas del mundo –veremos más sobre ello en el capítulo siguiente). El formato udf está igualmente soportado en Windows.

Para poder utilizar sistemas de ficheros formateados como ext2 se pueden utilizar herramientas especiales *no* propias de Windows, como por ejemplo LTOOLS, las cuales permiten leer y escribir datos en particiones ext2, ext3 o ReiserFS: `http://freshmeat.net/projects/ltools/`.

Linux soporta todos los tipos anteriormente mencionados, suponiendo que haya sido adecuadamente configurado para ello. Básicamente esta "configuración" consiste en tener preparados (compilados) los módulos necesarios para que puedan ser cargados por el *kernel*[22] de Linux.

[20] Tiene prestaciones más similares a las que proporciona ReiserFS.
[21] Al menos en la partición donde se va a instalar el sistema y el software.
[22] Núcleo del S.O.

Cuando añades un disco duro a tu ordenador, el disco está *virgen*, es decir, no tiene formato alguno. Algunos dispositivos (como el CD-ROM) constituyen una excepción ya que vienen con su propio formato de sistema de ficheros (estándar) que todos los S.O. reconocen. La primera acción es, pues, *formatear* el disco con el formato adecuado para el sistema de ficheros que deseemos. Windows detecta automáticamente que un disco no tiene formato, pero sólo permite formatearlo para los sistemas de ficheros de Windows (FAT y NTFS). Ciertas herramientas del ordenador permiten configurar los discos y muchos otros elementos del ordenador, tanto en Linux como en Windows.

2.7. CONFIGURACIÓN DEL ORDENADOR

Los entornos gráficos permiten configurar *ciertos aspectos* de funcionamiento del sistema con relativa facilidad y por usuarios relativamente inexpertos. Sin embargo, algunas características sólo se pueden configurar correctamente *cuando se sabe lo que se está haciendo*.

En muchas ocasiones, configurar adecuadamente ciertos servicios y/o componentes del sistema implica tener que leerse documentación, manuales, etc., y tener que editar determinados ficheros de configuración *desde la consola*.

2.7.1. Configuración de Windows

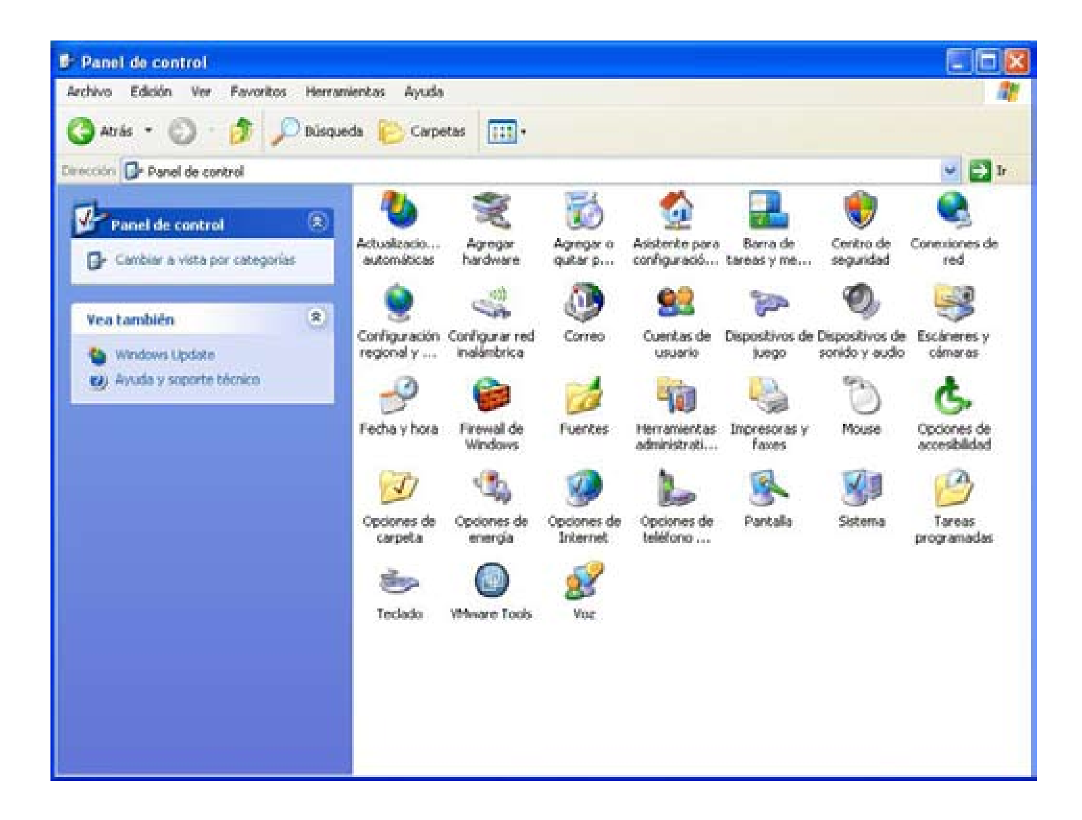


Figura 2.26. El panel de control de Windows

Windows facilita la configuración a través de un *panel de control*: un conjunto de utilidades de configuración agrupadas (véase la Figura 2.26). Se accede a él a través

del menú *Inicio* (esquina inferior izquierda) o también desde el icono *Mi PC*. En ambos casos aparecerán una serie de iconos con los que podemos configurar diversos aspectos de Windows. Muchas de estas utilidades se encuentran con el mismo nombre o similar en distintas versiones: 98, Me, NT, 2000 y XP. Hay herramientas en el panel de control para:

- *Actualizaciones automáticas*: permite configurar las opciones para instalación de parches de seguridad, Service Packs, etc.
- *Agregar hardware*: permite indicar a Windows que hemos añadido una tarjeta de expansión, un dispositivo USB, etc[23]. Sólo es necesario recurrir a él si no se produjo una detección automática al arrancar. Si tenemos "conflictos" entre dispositivos (véase el capítulo 4), hay que utilizar el panel *Sistema*. Siempre que vayas a agregar un nuevo hardware no detectado por el sistema, convendría que usaras el disquete o CD proporcionado por el fabricante.
- *Agregar/quitar programas*: permite instalar y desinstalar programas. Como los programas suelen venir con sus instaladores propios, este elemento se usa casi exclusivamente para desinstalar programas. Es recomendable, si el programa fue correctamente instalado, eliminarlo desde aquí para asegurarse de que se realizan los cambios necesarios. Igualmente podemos añadir o eliminar componentes optativos de Windows como juegos, controladores de fax o accesorios multimedia.
- *Configuración regional*: adapta características del ordenador al español o catalán (idioma, sistema monetario...). En general, permite cambiar una serie de características que dependen del país y la lengua que utilicemos. Las dos alfabetizaciones que aparecen junto al español son la tradicional (ch y ll son consideradas letras) y la moderna (que no las considera como tal).
- *Administración de energía*: controla el ACPI (gestión de energía moderna que permite incluso "hibernar" equipos). Mediante esta utilidad se puede hacer que los dispositivos (típicamente los discos duros y monitores) que no estén siendo utilizados durante un tiempo pasen a un modo de bajo consumo de energía. Con equipos portátiles aparecen más opciones (batería, configuración del consumo de energía del equipo, etc.) y permite "dormir" al ordenador, el monitor o el disco duro tras cierta inactividad. También se puede configurar el modo de "hibernación" del equipo.
- *Fecha y hora*: permite ajustar la zona horaria, fecha y hora y si se quiere que se realice automáticamente el cambio de horario verano/invierno.
- *Fuentes*: permite instalar/desinstalar tipografías y también modificar algunas de sus características. En XP se presenta como una carpeta "especial".
- *Impresoras y faxes*: permite acceder a cada impresora/fax conectada y configurar algunas características (papel, calidad de impresión, etc.), ver la lista de tareas de impresión pendientes o configurar una como predeterminada.

[23]Es decir, que tenemos un nuevo dispositivo instalado en el sistema.

- *Opciones de Internet*: controla aspectos de la presentación de páginas, modo de conexión; permite indicar qué software usaremos para correo, navegación, etc.
- *Opciones de teléfono y módem*: configura el módem, modo de marcado, tipo de conexión, etc.
- *Mouse*: permite cambiar diversas características del comportamiento del ratón: control de los botones para personas diestras o zurdas; velocidad del doble clic; forma del puntero (cursor) en pantalla; velocidad de movimiento; y otras dependientes del dispositivo utilizado.
- *Dispositivos de sonido y audio*: permite configurar el manejo de diversos aspectos del audio en el sistema.
- *Opciones de accesibilidad*: adapta características para facilitar el uso a discapacitados.
- *Pantalla*: permite cambiar las propiedades de la pantalla:
  - Escritorio: para cambiar la imagen de fondo del escritorio.
  - Protector de pantalla: son pequeños programas que evitan que el monitor presente siempre la misma imagen cuando, durante un tiempo, no hay interacción por parte del usuario.
  - Apariencia: permite cambiar características de los distintos elementos de las ventanas.
  - Configuración: permite cambiar la resolución y los colores de la pantalla.
  - Temas: efectos visuales en ventanas e iconos y permite cambiar iconos para determinados objetos (papelera, etc.).
- *Conexiones de Red*: permite acceder a las conexiones (interfaces) de red utilizadas, configurarlas, controlando los dispositivos empleados y permite introducir algunos datos necesarios (servidor de nombres, etc.) para cada interfaz por separado. Con ello configuramos la conexión a Internet (o a una "red local").
- *Sistema*: muestra información del ordenador y permite resolver posibles conflictos entre dispositivos. Presenta las siguientes opciones:
  - General: presenta datos de la versión del sistema Windows y del ordenador.
  - Hardware: permite el acceso, entre otros, a
    - Administrador de dispositivos: permite ver los dispositivos conectados, sus propiedades y posibles conflictos.
    - Perfiles de hardware: permiten habilitar y deshabilitar selectivamente distintos dispositivos al comenzar la sesión. Para ello se crea un nombre de perfil y se determina qué dispositivos estarán disponibles con él.
  - Opciones avanzadas: permite acceso a Rendimiento para configurar ciertos aspectos que influyen en el rendimiento del sistema.

- *Escáneres y cámaras*: permite configurar dispositivos de vídeo en el sistema y gestionar dispositivos de imagen (escáneres y cámaras digitales, p.e.).
- *Teclado*: permite configurar el teclado, en aspectos como la velocidad de repetición de las teclas o el tipo del teclado.

Sobre la barra de tareas de Windows:

Si no está bloqueada podemos cambiar su posición haciendo clic en alguna zona libre y, sin soltar el botón del ratón, arrastrándola a la posición donde queramos dejarla.

Si lo estuviese, para desbloquearla, accede al menú contextual de la barra haciendo clic con el botón derecho del ratón en una zona despejada de la misma y *desmarca* la opción *Bloquear la barra de tareas*.

Para cambiar el tamaño, nos movemos al borde y actuamos como lo haríamos con cualquier ventana. Además podemos cambiar sus propiedades desde la opción *Propiedades* de su menú contextual: cambiar cuándo se ve y cuándo no, la presencia del reloj y el tamaño de los iconos. También podemos cambiar los programas que aparecen en el menú de inicio.

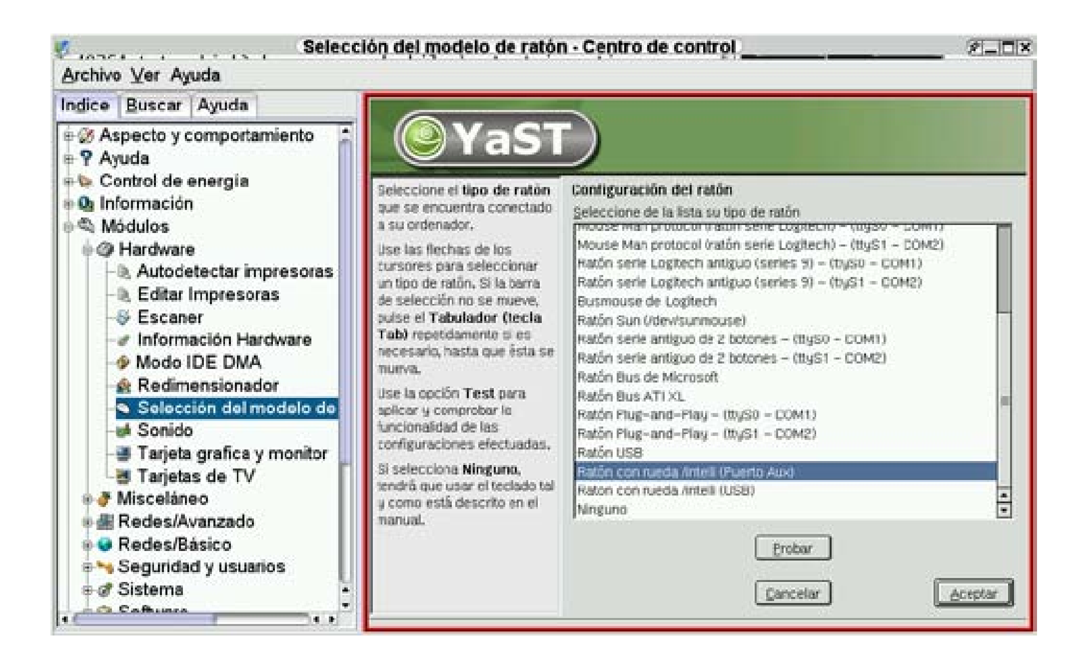


Figura 2.27. El centro de control de KDE en SuSE

2.7.2. Configuración de Linux

La configuración depende del entorno de usuario concreto que usemos. En KDE, por ejemplo, hay un panel de control similar al de Windows (véase la Figura 2.27).

Para poder configurar un sistema Linux debes ser administrador (*root*) del sistema, es decir conectarte al sistema como superusuario.

2.8. MÁS SOBRE LINUX

Linux es un sistema operativo completamente diferente de Microsoft Windows. Durante tus estudios tendrás que utilizar Linux para realizar diferentes trabajos. ¿Por qué usar Linux, si ya sabes usar Windows?

- Linux es un sistema operativo de la familia *Unix*. Unix cuenta con más de tres décadas de desarrollo e infinidad de utilidades disponibles y sigue muchos estándares para redes, comunicación, representación de datos, etc.
- Hay muchas herramientas (gratuitas) para *programación* que se consideran estándar *de facto*.
- También dispone de *entornos de usuario* que facilitan su uso: KDE y GNOME por ejemplo.
- Linux es *software libre* y, además, *gratuito* (para conocer bien lo que esto supone visita la dirección `http://www.gnu.org/philosophy/free-sw.es.html`). Lo primero es muy importante porque eso significa que disponemos del código fuente del programa para su estudio y posible modificación.

Hay una razón todavía más importante, utilizando Linux, si lo deseas, puedes aprender muchas cosas sobre informática ya que es un *sistema abierto* donde no se oculta nada[24]. Existe mucha documentación, manuales, etc. Con Linux podrás estudiar y aprender y, lo más importante, observar la aplicación práctica con todo lujo de detalles de muchos de los conceptos que estudiarás en la carrera.

¿Sabías que. . . ?

Linux es un sistema que fue creado por Linus Torvalds y posteriormente desarrollado con la ayuda de multitud de personas (*hackers*).

El objetivo que se perseguía era crear un sistema UNIX completo (que fuese POSIX), incluyendo: multitarea real, memoria virtual, bibliotecas compartidas, redes con conexión TCP/IP. . . Aunque inicialmente se desarrolló para la familia x86 de Intel (desde el 386 en adelante), existen ya versiones (algunas experimentales) para máquinas como Alpha (Digital), Sparc (SUN), 68000 (Mac), MIPS, Power PC y otras.

Tiene muchas de las características de los UNIX comerciales pero se distribuye bajo licencia GNU. En realidad, no forma parte del proyecto GNU (`http://www.gnu.org`), pero muchas de las utilidades que lo acompañan sí.

continúa en la página siguiente . . .

[24] A diferencia de Windows que es un sistema cerrado.

. . . viene de la página anterior

El proyecto GNU pretende crear un entorno de trabajo para ordenador empleando software 100 % libre. Ello incluye un sistema operativo GNU (*Hurd*) actualmente en desarrollo (ya hay una versión operativa de este sistema). Ya existen distribuciones totalmente GNU que incluyen su kernel propio, Hurd.

Probablemente, al principio, te basarás mucho en la utilización de herramientas gráficas de configuración, pero conforme vayas aprendiendo y cojas soltura, te atreverás a modificar los ficheros de configuración por ti mismo, sin la ayuda de otras herramientas. Ésta es una de las ventajas que proporciona la versatilidad de Linux: si quieres, puedes "abstraerte" de los detalles de "bajo nivel" como en Windows. Pero, *a diferencia de Windows*, si quieres o necesitas "arremangarte y coger el destornillador" en Linux puedes hacerlo (y encontrarás mucha documentación que te ayudará en tu trabajo).

Propiamente hablando, Linux es exclusivamente el kernel. Existen varias ramas de desarrollo (2.0, 2.2, 2.4, 2.5, 2.6) y distintas versiones dentro de cada rama (p.e., la última versión disponible en la rama 2.6 es la 2.6.21[25]). Las ramas impares (p.e. 2.5) se asignan a versiones *inestables* y experimentales (en desarrollo), mientras que las pares (p.e. 2.4) son versiones *estables*.

Hay gran cantidad de programas libres, *shareware*, e incluso comerciales, que se pueden utilizar con el kernel: compiladores, entornos de ventanas, editores, procesadores de texto, hojas de cálculo, etc. Reunir estos programas supone una gran inversión de tiempo y puede requerir bastante esfuerzo hacer que funcionen de manera conjunta.

Para que sea más fácil montar un sistema basado en Linux, se han creado las *distribuciones*, que reúnen un kernel (o varios, siempre versiones *estables*, posiblemente incluyendo ramas distintas) y una serie de programas que permiten tener una máquina lista para las tareas más habituales. Además, incluyen utilidades que facilitan las tareas de configuración y administración de los programas que se instalan.

A continuación se muestran algunas de las distribuciones *principales* más conocidas. Esta información se ha sacado del *"Rincón de Linux. Linux para hispanohablantes"*, `http://www.linux-es.org/`, en el apartado distribuciones:

RedHat: tiene muy buena calidad en contenidos y soporte a los usuarios por parte de la empresa que la distribuye. Fácil de instalar. Es la distribución estándar de Linux en EE.UU., `http://www.redhat.com`. Ahora se dedica sólo a empresas (y resulta muy cara para el usuario de a pie). El colectivo de usuarios de RedHat, patrocinado por la empresa, continúa con la versión para particulares (se puede descargar por Internet de forma gratuita): *Fedora*, `http://fedora.redhat.com`.

Debian: no la produce una empresa, sino una comunidad de voluntarios. Tal vez sea la mejor distribución de Linux que existe en la actualidad. Si tienes una buena conexión, podrás actualizarte fácilmente vía Internet. El proceso de instalación

[25]En el momento de redactar este capítulo.

es un poco más complicado, aunque se aprende mucho. Se compensa de sobra con la facilidad de actualización que proporciona, muy superior al de otras distribuciones. No se recomienda para no iniciados, aunque si quieres aprender de verdad y tienes tiempo que invertir, atrévete con ella. Mirror en España: `http://www.es.debian.org/`. Hay otras distribuciones, más amigables, basadas en ella y que comienzan a gozar de gran popularidad y prestigio, tales como **Ubuntu** (o **KUbuntu**, que es la versión que lleva el escritorio KDE, ya que la original usa Gnome) o **Knoppix**.

S.u.S.E: es una distribución que entró con mucha fuerza en el mercado español; siendo una de las más populares en Europa. La creó una empresa alemana, que hace unos años fue adquirida por *Novell* y que es la empresa que ahora le da soporte. Es fácil de instalar, ya que tiene un programa de instalación muy bueno. La rama profesional (para empresas) la podemos encontrar en `http://www.novell.com/linux/`. La versión para usuarios particulares se denomina OpenSuSE y la mantiene una comunidad de usuarios con el apoyo de Novell: `http://www.opensuse.org/`.

Mandriva: surgió de la reciente fusión de **Mandrake** y **Conectiva Linux**. Es otra de las más populares en Europa. Fácil de instalar. No obstante, resulta algo menos completa que SuSE. `http://www.mandriva.com/`.

Slackware: de las primeras. Ha pasado por un periodo de "crisis", durante el cual no se ha actualizado muy a menudo, aunque parece que está volviendo con fuerza. `http://www.slackware.com`.

Las empresas que hacen las distribuciones suelen cobrar por sus servicios (la excepción es Debian, que solamente cobra un donativo, aparte del soporte en CD). El precio suele ser muy asequible y realmente pagas por los "extras" que ofrecen:

- manuales,
- ayuda técnica,
- software que facilita la instalación.

Aunque estas empresas cobren por la distribución, es perfectamente *legal* efectuar copias para uso personal. De hecho, algunas distribuciones permiten descargarse las "imágenes" de los CD vía Internet, para que uno mismo pueda grabarse los CD de instalación. Por contra, si haces copias de Windows, cometes un acto *ilegal* y vulneras los derechos de propiedad intelectual de la empresa Microsoft.

Las distribuciones suelen ofrecer mecanismos para controlar qué está instalado y qué no. La manera más normal es agrupar los programas y posible documentación asociada en *paquetes* y prever mecanismos para instalar o eliminar dichos paquetes. De esta manera se puede tener un cierto control sobre lo que hay instalado y permitir que sea más fácil actualizar las versiones.

En el momento de la instalación, se suelen elegir los paquetes agrupados según alguna categoría adecuada (p.e. para uso en portátil, en estación de desarrollo de programas o para uso como servidor de red). Después, se pueden ir añadiendo y quitando

paquetes mediante órdenes especiales o entornos más o menos cuidados. Cada distribución suele incorporar mecanismos (uno o varios programas, gráficos o no) para la gestión de paquetes desde varios dispositivos (incluso desde Internet). Hay, fundamentalmente, dos *formatos de paquetes* para distribuir software en Linux "listo para funcionar":

RPM: sistema usado por RedHat, Mandriva y SuSE. Un paquete de RedHat, por ejemplo, *suele* funcionar sin problemas en SuSE.

deb: sistema usado por Debian. Es un sistema con mejoras importantes, superior a RPM, pero en principio incompatible con éste.

¿RPM y deb incompatibles?

Aunque en principio puedan parecer formatos de distribución de software incompatibles, en realidad existe una herramienta que permite convertir paquetes de software "en formato" RPM a deb y viceversa. Esta herramienta es el programa `alien`. Si quieres saber más, abre una consola, teclea `man alien` y pulsa Enter.

2.8.1. Sobre la instalación de Linux

Aunque suele variar entre distribuciones, la instalación tiene los siguientes "grandes pasos":

- Arrancar un sistema básico. Generalmente tenemos tres opciones:
  - Utilizar uno o más disquetes (y realizar la instalación del resto del software por Internet).
  - Utilizar el propio CD-ROM (el CD1 de la instalación) como dispositivo de arranque (si éste lo permite).
  - Utilizar el DVD de instalación como dispositivo de arranque (si éste lo permite).

Un paso previo a considerar

En muchas ocasiones, puede resultar conveniente ajustar y/o cambiar ciertas opciones de configuración de la BIOS (consulta los contenidos del capítulo 4, que versa sobre hardware):

Si instalamos desde CD-ROM o DVD (autoarrancable), hay que elegir el orden adecuado de arranque en el ordenador, es decir, indicar en la BIOS que se desea poder arrancar desde CD-ROM (o DVD) antes que del disco duro.

continúa en la página siguiente ...

... viene de la página anterior

Poner el máximo de memoria RAM posible como *extendida*. Linux requiere memoria extendida y *no* puede manejar memoria *expandida*.

Deshabilitar las opciones antivirus de la BIOS. No suelen ser compatibles con Linux. En cualquier caso, y gracias al sistema de permisos de los ficheros y el mecanismo de memoria protegida de Linux, los virus son casi desconocidos.

Si la placa proporciona algo como *shadow RAM* o *BIOS caching*, hay que deshabilitarlo en la configuración de la BIOS. Este tipo de uso de la memoria puede interferir en el mecanismo de operación de Linux con los dispositivos, bastante eficaz por "naturaleza".

Deshabilitar las opciones de *Advanced Power Management* (APM). Linux puede hacer un mejor control de la administración de la energía que cualquier BIOS.

Si se tiene RAM con detección de errores (soporte de bits de paridad en la RAM) y la placa base es capaz de manejarla (nuevamente consúltense los contenidos del capítulo 4), habilitar en la BIOS la(s) opción(es) que permiten disparar la interrupción correspondiente cuando se detecte un error en la memoria.

Deshabilitar opciones que digan algo así como "15-16 MB *memory hole*" o algo parecido. Esto puede entorpecer la gestión de la memoria en Linux.

- Utilizar el sistema básico para realizar las particiones y preparar el disco.
- Instalar los paquetes (aplicaciones) deseados.
- Configurar algunos aspectos (contraseña del superusuario, crear usuarios "normales", configurar las X, etc.).

Arranque del sistema básico

Para comenzar a hacer la instalación, es necesario tener un sistema Linux ya ejecutándose. Para conseguirlo, hay dos opciones principales.

Una posibilidad es crear uno o más disquetes de arranque con un sistema Linux preparado para el hardware más habitual y una serie de disquetes que tengan las utilidades básicas para configurar el sistema.

Estos disquetes se preparan a partir de ficheros especiales (*imágenes*) que se escriben en el disco utilizando el programa MS-DOS `rawrite` (se puede descargar de Internet). Este programa sirve para hacer copias *binarias* (por bloques o sectores) de ficheros.

No obstante es preferible usar la orden `dd` si se tiene acceso a un Unix (o Linux). Cada imagen tiene el tamaño justo del disco (excepto posiblemente la última) y en ella está su propio sistema de ficheros.

Una alternativa mucho más cómoda a los disquetes es la utilización de un CD-ROM (o DVD) como disco de arranque que ejecuta automáticamente un programa de instalación que incluye un sistema Linux básico.

continúa en la página siguiente ...

. . . viene de la página anterior

Para esto, se tendrá que cambiar la configuración de la BIOS de modo que pruebe el arranque desde el lector de CD (o DVD) antes que el disco duro.

También es necesario que el CD-ROM (o DVD) de la distribución esté ya preparado para ser utilizado como disco de arranque. Cualquier distribución se puede conseguir en CD-ROM (o DVD), de los cuales el primero se puede utilizar como arranque.

Usualmente instalas Linux sobre un ordenador que lleva Windows preinstalado. Puede ocurrir lo siguiente:

- Si sólo tienes un disco duro, has de hacer espacio para Linux. Ello implica *particionar* el disco y *redimensionar* el sistema de ficheros de Windows. Los discos duros pueden ser divididos en distintas partes (*particiones*), permitiendo que el ordenador las vea como si fueran discos distintos. Esto permite, por ejemplo, organizar mejor los datos dentro del disco o tener simultáneamente más de un sistema operativo en el ordenador. Para instalar Linux, será necesario particionar el disco de modo que pueda coexistir con MS-DOS, Windows u otros sistemas que puedas tener instalados. Necesitas al menos una partición para Linux y otra para Windows.
- Si tienes más de un disco, puedes destinar un disco a cada sistema operativo. Aunque sólo vayas a tener Linux en un disco determinado, también es aconsejable hacer diversas particiones puesto que necesitarás al menos una partición de *swap*.

Es aconsejable tener, al menos, las siguientes particiones en Linux:

- Una partición para el directorio raíz (/), esto es, el sistema de ficheros principal. Esta partición es absolutamente necesaria para que el sistema pueda funcionar. Su tamaño dependerá de la cantidad de disco que hayamos asignado a Linux, pero lo razonable sería al menos 4 GB (conviene reservar entre 6 y 8 GB) para una instalación con numerosos paquetes de software. No obstante, dependiendo de la cantidad de programas que vayamos a instalar en Linux, incluso podríamos necesitar más.
- Una partición para *swap*. Se utiliza para la memoria virtual. Su tamaño dependerá de la cantidad de memoria real que se tenga. Una regla útil es tener como mínimo el mismo tamaño que la memoria real y como mucho el doble. Un espacio superior a los 700 u 800 MB normalmente resultará excesivo (incluso aunque tengamos 512 MB o más de memoria RAM).
- Una partición para los datos de usuario (el directorio `/home`). El tenerla separada de la partición del sistema de ficheros principal (/) permite que sea más fácil cambiar de distribución o hacer copias de seguridad. Si lo vas a usar para prácticas, entre 4 y 5 GB es suficiente. Si lo vas a usar para aplicaciones multimedia, reserva mucho más.

¿Sabías que…?

Los requisitos mínimos teóricos para instalar Linux en un PC son muy básicos: procesador 386 o superior, 4 MB de memoria RAM y disquetera.

En la práctica, para poder hacer algo útil, las distribuciones suelen necesitar algo más:

32 MB sería el mínimo aceptable y 64 MB lo aconsejable para utilizar las X –servidor de gráficos– y disco duro con un mínimo de unos 600 MB (variará según lo que instalemos, es decir según distribuciones y el tipo de instalación que se haga).

Prácticamente cualquier placa base funciona. Se puede utilizar prácticamente cualquier unidad de CD-ROM (aunque no es necesario, es muy recomendable instalar desde CD-ROM).

Vamos a comentar algunas cosas más acerca de las particiones en un disco duro. La información acerca de las "partes" en las que se ha dividido el disco duro se guarda en la *tabla de particiones*, la cual se almacena en el sector 0 del disco. No puedes crear cuantas particiones quieras. Hay una serie de restricciones:

- En el sector 0 del disco sólo hay espacio para almacenar información relativa a 4 particiones, llamadas *primarias*.
- Una de las primarias puede ser una partición *extendida*, esto es, una "caja" que mantiene la descripción de una serie de particiones que se llaman *lógicas*.
- A las 4 particiones primarias, hayan sido o no definidas, se les asignan los números del 1 al 4. Las particiones lógicas se numeran del 5 en adelante.

Cuando instalas Linux, arrancas de un CD-ROM. El CD-ROM entra rápidamente en el modo de particionado y puede que te sugiera una distribución de particiones razonable. En cualquier caso debes saber que efectuar particiones es una acción potencialmente peligrosa. Antes de instalar Linux, haz una copia de seguridad de la información que tengas en Windows.

Más sobre particiones

El esquema de particiones descrito es válido *sólo* para la arquitectura PC. *Distintas* arquitecturas presentan formas *diferentes* de particionar los discos. En discos para sistemas BSD/SUN sólo se pueden hacer 8 particiones, siendo la tercera un caso "especial"; los discos para sistemas IRIX/SGI admiten un máximo de 16 particiones, siendo la novena y la undécima casos "especiales".

Las particiones pueden realizarse con el programa `fdisk`. Es aconsejable utilizar el de Linux que es más completo y manejable y no el de MS-DOS, que no entiende que existen otros sistemas operativos. No obstante, las distribuciones suelen ofrecer otras alternativas (programas en modo gráfico) que funcionan bastante bien.

continúa en la página siguiente …

... viene de la página anterior

PartitionMagic es un programa muy interesante, aunque es comercial. Este programa es capaz de crear, redimensionar, dividir, unir, recuperar y cambiar de tipo particiones del disco duro sin destruir datos. También es capaz de gestionar el arranque de varios sistemas operativos (Linux incluido).

A destacar: interfaz gráfica, lo cual permite gran facilidad de uso, soporte para discos de más de 20 GB, soporte de ext2 (sistema de ficheros de Linux) y potentes herramientas para chequeo de discos. Existen versiones para Windows 95/98, Windows 2000, Windows Me, Windows NT 4.0. y Windows XP.

También desde el "rincón" del software libre existen alternativas interesantes, aunque no tan desarrolladas. Concretamente, hay un programa llamado Parted que es de GNU. Este programa permite crear, destruir, redimensionar, mover y copiar particiones. Es útil para dar espacio a nuevos sistemas operativos, reorganizar el disco y copiar datos a nuevos discos. Recientemente se han desarrollado programas que ofrecen una intuitiva interfaz gráfica para manejarlo (como GParted o QtParted).

En cualquier caso, es mejor, si se tiene la oportunidad, preparar el espacio antes de instalar nada.

Los usuarios

Para añadir usuarios al sistema hay distintas opciones (y siempre ejecutándolas como root).

Editar directamente los ficheros /etc/passwd y /etc/group y crear manualmente las entradas y directorios correspondientes. No se recomienda a no ser que se sepa muy bien lo que se está haciendo.

Otra es utilizar órdenes especificas. Para añadirlos: adduser, addgroup. Para eliminarlos: userdel, groupdel.

Finalmente, es posible emplear programas como webmin (que permite gestionar cómodamente el sistema a través de una serie de páginas web, mediante tu navegador favorito) o los programas gráficos de gestión de usuarios que proporcionan las distribuciones, los cuales permiten utilizar ventanas y menús.

Durante el proceso de instalación se configuran una serie de aspectos del sistema. Estas configuraciones pueden hacerse antes, durante o después del proceso de instalación de paquetes. Algunas de las configuraciones más típicas incluyen:

- Elegir un password para el superusuario, o root.
- Dar de alta a un usuario en el sistema para el trabajo "habitual" (no es bueno, ni debe hacerse por motivos de seguridad, trabajar siempre como root).
- Detectar y configurar hardware (tarjeta de sonido, tarjeta ethernet, etc.)

- Configurar el acceso a Internet.
- Configurar la interfaz gráfica de usuario (detección de la tarjeta gráfica, configuración del servidor X).
- Preparar adecuadamente los cargadores LILO (LInux LOader) o GRUB (GRand Unifier Bootloader) para el arranque de Linux (y posiblemente Windows).

Vamos a hablar con más detalle sobre el gestor de arranque LILO. Debes saber que Linux puede convivir tranquilamente con otros sistemas operativos. Existen distintas maneras de conseguir esta convivencia, una de las más cómodas es utilizar *LILO* (LInux LOader). Cuando Linux se instala con éxito, suele permitir la selección del sistema operativo en el arranque mediante este programa. La idea básica es que LILO se carga al principio, dando al usuario la oportunidad de decidir cuál de los sistemas presentes en la máquina quiere arrancar. Además, LILO es capaz de arrancar sistemas operativos que no estén necesariamente en una *partición primaria*.

Más sobre LILO

LILO tiene diversas opciones que permiten fijar cuánto tiempo se espera para que el usuario decida qué quiere arrancar o qué parámetros se pasan a un sistema determinado (Linux admite que se le den ciertas opciones al arrancar).

Para configurarlo se utiliza el fichero `/etc/lilo.conf` o algún programa de control; aunque resulta conveniente leerse la documentación en `/usr/doc/lilo` y ejecutar `man lilo` y `man lilo.conf`.

Un problema que tiene LILO es que utiliza la BIOS para leer la geometría del disco, lo que con BIOS antiguas y discos grandes puede dar problemas. Por eso se aconseja que, si tienes un ordenador antiguo (más de ocho años), las particiones de arranque (las que contienen los ejecutables de los sistemas operativos) estén dentro de los primeros 1024 cilindros (8 GB). Con ordenadores recientes (de 7 años para acá) esto no ocurre ya que tienen BIOS modernas.

Algunas veces los problemas de LILO con los ordenadores antiguos pueden resolverse fijando el modo *LBA* (Linear Block Access) de acceso a disco en la configuración de la BIOS y/o usando la opción *linear* en la configuración de LILO.

En la actualidad se recomienda usar la opción *lba32* en sistemas con BIOS “modernas” (equipos comprados con posterioridad a 1998). Esto permite arrancar sistemas instalados en particiones más allá del cilindro 1024.

Ten en cuenta que, normalmente, desde Linux podrás acceder al contenido de los discos y particiones Windows. Incluso en el caso del sistema NTFS puedes leer y escribir ficheros sin ningún problema *si usas las herramientas adecuadas* (por ejemplo, el *driver* NTFS-3G del proyecto Linux-NTFS).

Coexistencia de los sistemas operativos

Linux no tiene problema en coexistir con otros sistemas operativos. Sin embargo son interesantes algunas observaciones.

Esta buena vecindad no siempre es correspondida. Por ejemplo, algunas versiones de Windows cuando se instalan asumen que son los únicos sistemas; ello significa que escriben en el sector de arranque del disco "sin mirar" si había algo antes (o sin importarles que se pueda perder lo que antes había). Además exigen estar en el disco primario. Solución: instalar primero Windows (y darle la *primera* partición del *primer* disco) y después Linux. No obstante, antes de instalar Windows, se deberían hacer las particiones, si no Windows se "queda" con todo el disco.

Linux entiende gran cantidad de sistemas de ficheros (p.e. FAT, FAT extendido, HFS o Hierarchical File System de Macintosh, NTFS, etc.).

Se pueden ejecutar otros sistemas en una "máquina virtual" utilizando `vmware`.

La máquina virtual es un software que *emula* una *computadora personal* (PC). Esto es, un ordenador con sus discos duros, su tarjeta de sonido, su unidad de CD-ROM, su disquetera, su memoria RAM, su tarjeta gráfica, su propia BIOS y, por supuesto, su *procesador*, incluyendo soporte completo (emulación) del juego de instrucciones x86 de Intel.

Podemos configurar varias máquinas virtuales (múltiples PC) dentro de nuestro ordenador *real*. Por supuesto, en cada uno de estos PC virtuales podemos instalar el S.O. que queramos.

El S.O. instalado y las aplicaciones que, sobre él, instalemos se ejecutarán *dentro* de la máquina virtual, usando el juego de instrucciones del procesador emulado.

La emulación del disco duro se realiza sobre un fichero. Este fichero irá creciendo *dinámicamente* hasta alcanzar el tamaño máximo establecido en la configuración de la correspondiente máquina virtual. Ello implica que todos los cambios se pueden deshacer fácilmente.

`vmware` es comercial, aunque existe una versión gratuita con menos características. Es necesario adquirir una licencia para poder usarlo. En `www.vmware.com` puedes descargarte versiones de prueba (30 días). Existen versiones para Windows y para Linux. Sin embargo, existe una alternativa desde el software libre, *aunque no tan completa y funcional*.

`bochs`, `bochs.sourceforge.net`. Es idéntico en cuanto a concepto a `vmware`, ya que es un emulador de PC (escrito en C++). Incluye emulación de los procesadores x86 de Intel (386, 486 o Pentium), soporte para dispositivos de entrada y salida comunes y una BIOS propia. Actualmente es capaz de ejecutar Linux, Windows 95, MS-DOS, Windows NT, Windows 2000 y Windows XP en la emulación. Existen versiones para Windows y para Linux.

Finalmente, una alternativa a LILO que resulta hoy en día la más usada (es el cargador por defecto en la mayoría de distribuciones y además es GNU :-): GRUB (GRand Unified Bootloader), un "cargador" que pretende unificar el proceso de arranque de los sistemas operativos en PC: `www.gnu.org/software/grub/`. Características más destacadas:

continúa en la página siguiente . . .

. . . viene de la página anterior

Carga de múltiples imágenes (interesante para sistemas *modulares* como *Hurd*).

Puede cargar Linux, Hurd, DOS, Windows, OS/2, etc.

Ofrece una interfaz amigable (menús) para gestionar la carga de sistemas operativos.

Soporta múltiples sistemas de ficheros.

Independencia de la geometría del disco que proporciona la BIOS. Si ésta soporta LBA y el disco puede ser accedido mediante esta técnica, no hay problema en arrancar imágenes situadas más allá de los 8 GB (cilindro 1024).

Arranque de imágenes desde red local.

Desde Windows *no* puedes ver el contenido de los discos y particiones Linux, a no ser que utilices programas específicos ajenos a Microsoft (como las LTOOLS, comentadas anteriormente). ¡Ojo! Si instalas Windows después de instalar Linux (cosa más frecuente de lo que uno quisiera), hay varios peligros potenciales:

- Windows puede reasignar particiones o instalarse en una partición Linux, destruyendo su contenido.
- Windows puede borrar a LILO o GRUB.

Por tanto, es conveniente que crees un disquete de arranque Linux antes de instalar Windows (el disquete de arranque se crea durante la instalación). En cualquier caso, asegúrate de haber hecho antes una copia de seguridad de los datos importantes que tengas en Linux. En resumen, el proceso de instalación de Linux sobre una máquina con Windows o de Windows sobre una máquina con Linux es *potencialmente peligroso*. Otras fuentes potenciales de problemas:

- Hardware muy reciente: Linux puede no disponer todavía de controladores.
- Ventanas (sistema X Window): algunas tarjetas de vídeo son problemáticas. Consulta si tu tarjeta está *plenamente* soportada en Linux.
- Dispositivos USB que no siguen los estándares (cosa frecuente en los módems USB).
- Módems WinMódem (son módems internos incompletos que no se ajustan a estándares).
- Dispositivos (módems ADSL, por ejemplo) que requieren de *drivers* (programas de control) específicos para funcionar y el fabricante sólo los desarrolla y mantiene para Windows.

APÉNDICES (no para examen)

Compilación del kernel

Entendemos por kernel el *núcleo* del sistema operativo, esto es, su parte principal. Normalmente, no tendremos que compilar el código fuente (las instrucciones del programa) del kernel, ya que el que viene ya compilado e incluido en las distribuciones suele estar preparado para gran cantidad de hardware distinto. Sin embargo, esto hace que tenga gran tamaño y no sea el más eficiente para nuestro sistema (aunque esto lo solucionan las distribuciones utilizando módulos con todas las partes del kernel que así lo permitan).

Puede suceder que adquiramos nuevo hardware, tengamos nuevas necesidades o consigamos una nueva versión del kernel, la cual deseamos instalar (por ejemplo, porque resuelve ciertos problemas de seguridad). En cualquier caso, pudiera ser que necesitásemos compilar el kernel. A la hora de compilar el kernel hay que tener bastante claro qué hardware tenemos y cómo queremos utilizarlo. Los pasos que hay que seguir son:

- Antes de nada leer la documentación, concretamente el `Kernel-HOWTO` (lo encontrarás dentro de algún subdirectorio de `/usr/share/doc/`).
- Configurar el kernel.
- Hacer la compilación propiamente dicha.
- Compilar e instalar los módulos.

Lógicamente, esto hay que hacerlo como superusuario y en el directorio correspondiente (`/usr/src/linux` suele ser un enlace simbólico al directorio de la versión que toca del kernel).

Configuración del kernel

La configuración del kernel consiste en ir decidiendo qué características queremos incluir en él, p.e., capacidad para manejar discos SCSI, impresoras en el puerto paralelo, controlar sistemas multiprocesador o incluir soporte para nuestra tarjeta de sonido. Hay tres opciones:

- `make config`: es la más primitiva. Va preguntando las opciones una a una. Hoy día prácticamente no se utiliza.
- `make menuconfig`: se utilizan menús en modo texto. Es bastante cómoda y más flexible que la anterior.
- `make xconfig`: la configuración se realiza con ayuda de ventanas y botones. Probablemente sea la más aconsejable.

Con cualquiera de estas maneras, se puede acceder a una ayuda que aconseja acerca de cada opción del kernel.

Módulos

Hay determinadas partes del kernel que se pueden configurar como módulos: partes que, bajo demanda, pueden cargarse dinámicamente en memoria y descargarse automáticamente mientras el kernel está ejecutándose. Algunas ventajas de utilizar módulos son:

- El kernel puede ser más pequeño y cargar sólo aquellos módulos que realmente utilice (si usamos LILO, el kernel no puede exceder de un cierto tamaño, por ejemplo).
- La carga y descarga de los módulos puede hacerse de forma automática.
- Determinados periféricos (p.e. SCSI) necesitan estar encendidos en el momento de inicializarse el controlador. Si se utilizan módulos es posible no tener los dispositivos encendidos desde el arranque.

Compilación

Una vez configurado el kernel, hay que realizar la compilación propiamente dicha. Los pasos necesarios son:

- `make dep`: con esto, el sistema encuentra las dependencias entre las distintas partes del kernel.
- `make clean`: esta orden limpia los ficheros que puedan quedar de compilaciones anteriores. Esto es fundamental para asegurar que estamos compilando todo lo que toca. Por ello, existen otras opciones "más fuertes", como `make mrproper` la cual "hace una limpieza más profunda"[26].

Finalmente creamos el fichero que contendrá el kernel. Las opciones son:

- `make zImage`/`make bzImage`: crea una imagen del kernel en un fichero, que después hay que colocar donde interese.
- `make zdisk`/`make bzdisk`: crea un disco de arranque con el nuevo kernel.
- `make zlilo`/`make bzlilo`: crea el nuevo kernel y ejecuta `lilo`.

La diferencia entre la forma `z*` y la `bz*` (ambas indican el tipo de compresión, ya que la imagen del kernel se guarda siempre comprimida) es que la segunda permite kernels de mayor tamaño, ya que el nivel de compresión es mayor.

Compilación e instalación de los módulos

Ésta es la parte final. Basta con:

- `make modules`: compila los módulos.
- `make modules_install`: copia los módulos en el sitio adecuado.

Después, es conveniente utilizar la orden `depmod -a` para que el sistema sepa qué relación tienen entre sí los distintos módulos.

[26] Esto *no* es una broma.

Fuentes de documentación

En este capítulo sólo hemos hecho una somera introducción a Linux. Existe una gran cantidad de aspectos que no hemos tratado y que se pueden encontrar en distintos sitios. En `/usr/share/doc` (o `/usr/doc`) existe una carpeta por cada uno de los paquetes instalados en el sistema con información sobre su uso, configuración, etc.

Muchos de los directorios se refieren a programas concretos, pero dentro de `/usr/share/doc` encontrarás un `FAQ` y un `HOWTO` (el lugar específico depende de la distribución), donde hay muchos documentos sobre Linux en general. Además existe mucha información en forma de *info* accesible, por ejemplo, desde Emacs.

En particular, recomendamos la utilización del programa `dwww` que permite *navegar* (y hacer búsquedas específicas) a través de la documentación del sistema usando Mozilla o Konqueror, por ejemplo. El único "inconveniente" es que requiere la instalación de un servidor `http` (Apache, p.e.), pero ten en cuenta que no hace falta tener una conexión permanente (ni siquiera temporal) a Internet para instalar en el ordenador un servidor web.

Existen muchos grupos de *news* relacionados con Linux: los de la jerarquía `comp.os.linux` y en español los de `es.comp.os.linux`. Indirectamente, también lo están los relacionados con GNU (`gnu.*`). También hay distintas *listas de correo* creadas por grupos de usuarios y muchas distribuciones tienen también sus propias listas de correo (p.e. Debian) y, sobre todo, *foros electrónicos*. A través de sus páginas web puedes consultarlas, suscribirte...

Entornos gráficos de trabajo

La interfaz gráfica de usuario de los sistemas con Linux se basa en el entorno Xwindow. Existen diversas implementaciones de las X, la más común en las distribuciones es la denominada X.Org, "bifurcación" de XFree86 (la usada generalmente hasta el año 2004, aproximadamente) en su versión 4.4.0. Ambas son versiones libres basadas en X11R6 (versión 11 de las X, la primera ampliamente aceptada y difundida, release 6).

Uno de los principales principios de diseño del sistema Xwindow, y del propio UNIX, es que la funcionalidad se consigue mediante la *cooperación* de componentes separados, en lugar de tener un único componente que haga todo el trabajo. La principal ventaja de esta filosofía es que una parte del sistema puede ser fácilmente cambiada simplemente reemplazando el componente adecuado. En el caso de UNIX, tenemos, por ejemplo, el intérprete de órdenes (fácilmente cambiable por otro programa distinto del que viene por defecto).

En el caso de las X, el mejor ejemplo de esto es el concepto de *gestor de ventanas*, el cual es esencialmente el componente que controla la apariencia de las ventanas y proporciona los medios para que el usuario pueda interactuar con ellas. Prácticamente todo lo que aparece en la pantalla con las X está dentro de una ventana y un gestor de ventanas simplemente las *maneja*. La gente es diferente y usa los ordenadores de formas diversas y para tareas distintas ¿Por qué pensar que todos tenemos que usar la misma interfaz? Afortunadamente, las X no padecen de esta rigidez y se han desarrollado un gran número de gestores de ventanas, los cuales proporcionan no sólo

diferentes aspectos, sino también diferentes “comportamientos”. Es más, muchos de estos programas son altamente configurables y adaptables a los deseos del usuario.

Más aún, hace unos años se desarrollaron los primeros *entornos de escritorio* (p.e. GNOME y KDE). Estos programas estarían un nivel por encima de los gestores de ventanas (de hecho, se encargan de manejarlos). Pretenden proporcionar una interfaz gráfica de trabajo mucho más completa, tanto con el sistema operativo, como con las diversas aplicaciones que podamos manejar. Además, proporcionan su propio conjunto de utilidades y aplicaciones integradas en el entorno de trabajo (mismo aspecto y forma de funcionar). El trabajo se reparte en tres niveles:

- el servidor X,
- el gestor de ventanas y
- el entorno de escritorio.

Servidores X

Los servidores X son una serie de programas que interactúan con la tarjeta de vídeo y ofrecen a los programas que están por encima las facilidades básicas para el manejo de ventanas, uso de tipos de letra, dibujo, generación de eventos de los dispositivos de entrada de datos (teclado, ratón...), etc. Normalmente, las distribuciones (como SuSE) incluyen programas que detectan la tarjeta y configuran el servidor X de forma adecuada. Por si acaso algo fallase, te ofrecemos esta pequeña guía para que puedas consultar material que ayude a resolver tu problema.

Según la tarjeta de vídeo que tengamos, tendremos que elegir uno u otro servidor. Para ello resulta conveniente mirar la documentación, `/usr/X11R6/lib/X11/Cards` (su ubicación puede variar en función de la distribución). Una versión más actualizada se puede encontrar en el *wiki* de X.Org, `wiki.x.org/wiki`, o bien en la web del proyecto XFree86, `www.xfree86.org`. Datos que son importantes conocer para su configuración: nombre, chipset y fabricante de la tarjeta y la memoria que ésta tiene. Por ejemplo: ATI Mach64 3D RAGE II, fabricante ATI, chipset ati, memoria 32 MB. Para más detalles: `manxorg.conf` o bien `manXF86Config`.

Una vez tenemos el servidor, es necesario configurarlo mediante el fichero `xorg.conf`, si usamos X.Org, (lo más normal) o bien `XF86Config` (versión 3.3.6) o `XF86Config-4` (versiones 4.*.*) si usamos XFree86. Ambos ficheros se encuentran en el directorio `/etc/X11`. Esto se puede hacer a mano o con programas de configuración específicos. Un programa que siempre “está”, aunque funciona en modo texto, es `Xorg-configure` o bien `xf86config`. Funciona respondiendo a una serie de preguntas y eligiendo las opciones adecuadas de entre todas las posibles.

También existen herramientas “gráficas”, aunque suelen ser específicas de las distribuciones: `xf86cfg`, en Mandrake, `XFDrake`, en SuSE, `SaX2` (muy buena). Es posible probar el efecto de la configuración “en directo” utilizando el programa `xvidtune`. Los servidores X suelen sacar bastante provecho de la tarjeta y, con un poco de *hacking*, podemos tener resoluciones y frecuencias de refresco muy aceptables (el documento denominado `XFree86-Video-Timings-HOWTO` es una lectura dura pero muy aconsejable).

Hace ya más de cinco años que las X sufrieron cambios considerables, igualmente "heredados" por la bifurcación actualmente usada: X.Org. Se pasó de la versión 3.3.6 a una nueva rama de desarrollo, la 4. Cuestiones importantes:

- La arquitectura ha sido completamente rediseñada en la versión 4.
- Introduce un sistema *modular*, donde se requiere especificar los módulos deseados en el fichero `xorg.conf` o bien `XF86Config-4`.
- Los módulos se cargan *dinámicamente* en memoria bajo demanda por medio del servidor X.
- Sólo hay un *único programa* que actúa como servidor X, independientemente de la tarjeta que tengamos (esto se ha unificado), en contraposición a la lista de programas *dependientes de la tarjeta* que podían actuar como servidores X en la versión 3.
- El nuevo servidor "unificado" soporta la mayoría de tarjetas modernas de la mayoría de fabricantes e incluye, además, la mayor parte de las tarjetas soportadas en la versión 3 (pero no *todas*, de ahí que aún hablemos en estos apuntes de la versión 3.3.6).
- Es por eso que alguna tarjeta soportada en la versión 3, no lo esté aún en la versión 4. Antes de hacer el cambio, debemos asegurarnos (pero sólo si tienes un PC muy antiguo).
- Información sobre los módulos y sus posibilidades: ejecuta las órdenes `man Xorg` y `man xorg.conf`, o bien ejecuta la orden `man XF86Config`, y consulta `www.x.org` o bien `www.xfree86.org`.
- Otra "puerta que se abre" en Linux con la versión 4 de las X es la posibilidad de usar la aceleración 3D de nuestra tarjeta (caso de que la tenga). Esto es útil para programas de CAD, multimedia (películas), juegos. . .
- Para ello debemos especificar en la sección `"Module"` del fichero `xorg.conf` (o bien `XF86Config-4`) las líneas `Load"glx"` y `Load"dri"` (cargar módulos GLX y DRI).
- Además hay que cargar el módulo del kernel (rama 2.4 o 2.6) de Linux específico para nuestra tarjeta (la aceleración la proporciona realmente el kernel). Existen fabricantes, como Nvidia, que proporcionan los *drivers* específicos para el kernel en su página web. La mayoría de distribuciones incluyen utilidades para descargarse e instalar en el sistema estos *drivers*.
- No todas las tarjetas 3D soportadas en la versión 4 tienen soporte para aceleración en el kernel. En algunos casos, hay que esperar. . .

Gestores de ventanas

Los manejadores de ventanas son los encargados de hacer cosas como decidir los bordes de las ventanas, manejar los eventos (teclas, ratón, etc.) que genera el servidor X, etc.

Hay bastantes manejadores; algunos comerciales (con cierto prestigio) como *Motif* y otros libres como *fvwm*. Cada uno de ellos tiene bastantes aspectos que se pueden personalizar. Entre los más populares: *fvwm*, *twm*, *Sawfish*, *Icewm*, *WindowMaker*, *AfterStep*, *Enlightenment*, ...

Entornos de escritorio

Se encargan de proporcionar la interfaz última de trabajo con el usuario. Vienen con una serie de programas de aspecto y comportamiento uniforme y coherente, con el objeto de facilitar el trabajo del usuario. Además, ofrecen, entre otras, herramientas de:

- Atención al terminal (*login*) en modo gráfico,
- Gestión de acciones "arrastrar y soltar" a nivel de escritorio y entre aplicaciones que las soporten,
- Barras de tareas, menús flotantes y contextuales e iconos en el escritorio,
- Acceso a menús, desde los cuales se pueden lanzar la mayoría de las aplicaciones instaladas en el sistema,
- Configuración de todos los aspectos del entorno (incluyendo la configuración del gestor de ventanas).
- Destacan GNOME y KDE (además, son software libre).

GNOME. El proyecto GNOME pretende construir un escritorio completo y amigable basado enteramente en software libre. Como gestor de ventanas suele usar *Sawfish* o *Metacity*. Está basado en las librerías GTK como las aplicaciones que incluye, ofreciendo de esta forma un aspecto y comportamiento coherente. GNOME forma parte del proyecto GNU.

KDE. Proporciona una interfaz consistente para las aplicaciones X, tanto en apariencia como en funcionalidad. KDE contiene un conjunto básico de aplicaciones tales como un gestor de ventanas (*kwm*), gestor de archivos, emulador de terminal, sistema de ayuda y configuración de la pantalla. KDE especifica un *toolkit GUI* estándar así como unas librerías gráficas, las QT, que usan todas las aplicaciones. Han publicado una guía de estilo que recomiendan seguir a todos los desarrolladores de aplicaciones KDE. Desde la versión 2 incluye navegador integrado en el sistema de ficheros (Konqueror, como Explorer en Windows) y un conjunto integrado de herramientas ofimáticas.

Direcciones de interés para Linux

Finalmente, podemos comentar que hay numerosas páginas *web* dedicadas a Linux. A continuación, damos algunas URL de interés. En castellano:

- `http://sunsite.rediris.es`: tiene Linux y otras cosas.
- Barrapunto, `http://barrapunto.com`: contiene noticias varias sobre Linux y software libre, secciones sobre temas específicos y acceso a URL relacionadas.
- Proyecto Lucas, `http://lucas.hispalinux.es`: según dicen, es la mayor biblioteca en español dedicada a GNU/Linux de todo el planeta. Lo que sí he comprobado es que hay mucha información, incluyendo manuales a todos los niveles y todo en castellano.
- `http://www.es.kernel.org/`: acceso a varias versiones del kernel de Linux. Es un *mirror* de Red Iris (la red pública que agrupa a todas las universidades españolas) para `www.kernel.org`.
- La mayor comunidad de Debian en español, `http://www.esdebian.org`: documentación, software, manuales, etc., relacionada con Debian.
- El portal hispano de Ubuntu: `http://www.ubuntu-es.org`.

En inglés tenemos:

- `http://www.linuxhq.com`: con información sobre el kernel.
- `http://www.linux.org`: un poco de todo.
- `http://www.tldp.org/`: un montón de documentación acerca de Linux y acceso a otras URL de interés.
- `http://mercury.chem.pitt.edu/~sasha/LinuxFocus/`: revista electrónica sobre Linux (versión en castellano disponible).
- `http://www.kde.org/`: página principal del proyecto KDE.
- `http://www.gnome.org/`: página principal del proyecto GNOME.
- `http://www.gnu.org/`: página principal del proyecto GNU.
- `http://www.linux.org/hardware/laptop.html`: cómo instalar Linux en un montón de portátiles.
- `http://www.linuxprinting.org/`: todo acerca de la impresión desde Linux.
- `http://www.linmodems.org/`: cómo hacer funcionar nuestro módem en Linux (si se puede).

- `http://freshmeat.net/`: para buscar software de todo tipo (y de lo más variopinto) para Linux. Si no sabes si un determinado programa de Windows existe en Linux, éste es un buen sitio para empezar a averiguarlo.
- `http://lwn.net/`: las últimas noticias del mundo Linux.
- `http://xwinman.org/`: información sobre gestores de ventanas y entornos de escritorio para X y acceso a URL relacionadas.

CAPÍTULO 3

CODIFICACIÓN DE LA INFORMACIÓN

3.1. INFORMACIÓN

La informática trata de la adquisición, representación, tratamiento y transmisión de la información. Para llevar a cabo automáticamente estas operaciones se utilizan unas máquinas llamadas ordenadores.

Informática es una palabra de origen francés, formada por la contracción de los vocablos **infor**mación y auto**mática**. La Real Academia Española de la Lengua define la informática como el conjunto de conocimientos científicos y de técnicas que hacen posible el *tratamiento automático de la información* por medio de los ordenadores. En la Figura 3.1 se muestra un esquema de lo que se debe entender por tratamiento automático de la información. La información consistente en un conjunto de datos de entrada, se somete a unas operaciones o transformaciones automáticas para producir unos resultados o datos de salida.

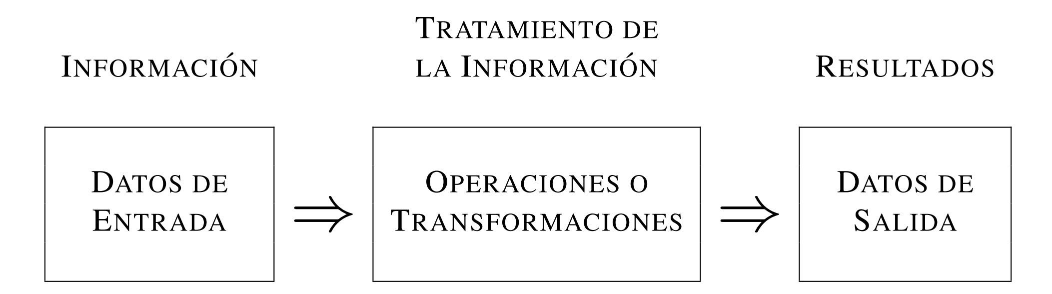


Figura 3.1. Esquema de tratamiento de la información

Algunos ejemplos de tratamiento de la información pueden ser:

- ordenar un conjunto de datos,
- dar formato a un texto,

- calcular estadísticas (media, desviación típica, moda, etc.) de un conjunto de valores numéricos,
- analizar el código de un programa escrito en C y producir el código máquina correspondiente, listo para ser ejecutado en un ordenador determinado.

Un ordenador, computador o computadora, es una máquina capaz de aceptar unos datos de entrada, efectuar con ellos operaciones aritméticas y lógicas, y proporcionar la información resultante a través de un medio de salida. Todo ello se realiza de forma automática bajo el control de un programa de instrucciones, previamente almacenado en el propio ordenador.

Los datos son conjuntos de símbolos utilizados para expresar o representar un valor numérico, un hecho, un objeto o una idea, y que se presentan en la forma adecuada para ser manipulados. Es decir, en informática el término "dato" tiene un significado muy amplio, y puede ser cosas tan diferentes como una temperatura, una altura, una matrícula de coche, una frase de un libro, una imagen, etc. Las preguntas que surgen inevitablemente son: *¿cómo representamos estos datos en el ordenador de forma adecuada?*, *¿cómo representamos la información en los ordenadores de manera eficiente (que facilite su diseño, operatoria y al mismo tiempo permita una fácil manipulación de la información)?*.

Una *codificación* es una transformación de los elementos de un conjunto en los de otro, de tal manera que a cada elemento del primer conjunto le corresponde un elemento distinto del segundo. Por tanto, *asocia representaciones* (signos o símbolos) *a los elementos de un conjunto* (datos o "significados"). Así pues, cualquier información que deseemos tratar automáticamente en un ordenador, representada mediante alguno de los códigos de comunicación habituales entre las personas (lenguaje, imágenes, sonidos, etc.), deberá tener asociado su código en la máquina (binario, como veremos) el cual deberá ser conocido por el ordenador y distinto del código máquina de cualquier otra información. En la entrada y en la salida del ordenador se efectúan automáticamente los cambios de código oportunos para que en el exterior la información sea directamente comprendida por los usuarios.

Veamos un par de ejemplos. En occidente codificamos los números con sucesiones de símbolos de un "alfabeto" (el conjunto de los dígitos) que tiene diez elementos: $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$.

Para representar los números naturales con este conjunto de dígitos, ponemos en correspondencia una sucesión de estos símbolos (dígitos) con un conjunto (infinito) de significados concretos: el conjunto de los números naturales.

$$\begin{array}{lll@{\qquad\qquad}lll} 1 & \rightarrow & \circ & 10 & \rightarrow & \circ\circ\circ\circ\circ\circ\circ\circ\circ\circ \\ 3 & \rightarrow & \circ\circ\circ & 15 & \rightarrow & \circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ\circ \end{array}$$

Imaginemos ahora que tenemos un alfabeto compuesto por estos dos símbolos:

$$\{A, B\}.$$

Queremos codificar el conjunto formado por estas diez letras griegas:

$$\{\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \theta, \kappa, \lambda, \omega\}.$$

Para ello, vamos a asociar una secuencia distinta formada por letras A y B a cada letra griega:

$$
\begin{array}{lllllll}
\alpha & \rightarrow & A & \quad & \zeta & \rightarrow & BBB \\
\beta & \rightarrow & B & & \theta & \rightarrow & ABAB \\
\gamma & \rightarrow & AA & & \kappa & \rightarrow & BABA \\
\delta & \rightarrow & BB & & \lambda & \rightarrow & AAAB \\
\epsilon & \rightarrow & AAA & & \omega & \rightarrow & BBBA
\end{array}
$$

3.2. UNOS Y CEROS

Los sistemas *digitales* son aquellos en los que los datos están representados mediante una magnitud física que varía de forma *discreta*. Un sistema digital en el que los valores posibles son sólo dos se llama *binario*. En la actualidad, *los ordenadores son digitales binarios*, y es por ello que utilizaremos como código máquina el sistema binario de codificación.

Si es preciso manipular datos analógicos (como el sonido, por ejemplo), se utilizan dispositivos periféricos adicionales que se comunican con el ordenador mediante conversores analógico-digitales (siguiendo con nuestro ejemplo, la tarjeta de sonido).

A la *unidad de información* empleada en los ordenadores se le conoce con el nombre de *bit* (“**bi**nary digi**t**”), que significa *dígito binario* y, por tanto, sólo puede tener uno de dos posibles valores (0 ó 1). ¿Cómo interpretamos el significado de un bit individual? Caben varias interpretaciones (todas aquellas que sólo requieran dos estados distintos). Veamos algunas de ellas:

- todo/nada,
- cierto/falso,
- sí/no,
- uno/cero,
- ying/yang.

La tecnología actual de los ordenadores hace que resulte fácil almacenar y manejar bits. ¿Por qué? Porque en un ordenador digital binario, los componentes más básicos son dispositivos electrónicos con dos estados estables (representados lógicamente mediante 0 y 1). De esta forma, todas las informaciones se representan mediante *códigos binarios*.

- El valor 1 puede consistir en que haya un cierto voltaje en un circuito y el valor 0 en la ausencia de dicho voltaje (o la presencia de un voltaje diferente).
- También pudiera ser que el valor 1 consista en que un condensador esté cargado y el valor 0, en que esté descargado.

- O bien, el valor 1 puede consistir en que la superficie de cierto material presente un agujero, y 0 en que es lisa.
- Hay infinidad de situaciones en "física" y "electrónica" que se corresponden con un escenario en el que sólo se puede dar una entre dos situaciones posibles, como acabamos de ver.

Además, existe un *valor añadido*: representar sólo símbolos de un alfabeto tan reducido *simplifica en gran medida el diseño* de ordenadores, *facilita su construcción* a gran escala (se abaratan mucho los costes, ya que los circuitos a utilizar son bastante simples) y los hace *más fiables* (es más fácil construir circuitos sencillos que no fallen).

Ten en cuenta que...

Los siguientes tres factores influyen a la hora de seleccionar una codificación: (i) el coste de la traducción, (ii) el coste del almacenamiento y (iii) el coste del tratamiento de la información.

3.2.1. Memoria

La *memoria principal* (memoria RAM, del inglés *Random Access Memory*) de un ordenador sirve para almacenar *temporalmente*[1] programas y datos. El procesador sólo puede ejecutar un programa cuando éste está almacenado en la memoria principal. Sin embargo, la memoria RAM tiene la desventaja de ser volátil y de tener una capacidad relativamente pequeña (en relación con dispositivos de almacenamiento como los discos duros).

¿Sabías que...?

La idea de aislar los componentes necesarios para construir una máquina de cálculo fue originalmente propuesta por Charles Babbage en 1833, más de un siglo antes de que se construyera el primer ordenador electrónico de propósito general.

En 1945, John von Neumann propuso utilizar la estructuración ideada por Babbage para construir ordenadores electrónicos, incorporándole dos nuevas ideas a su diseño: debían basarse en representaciones binarias (una idea original de Konrad Zuse) y debían almacenar los programas en memoria. La mayoría de los ordenadores actuales siguen estas directrices, conocidas como *arquitectura von Neumann*.

La memoria principal de un ordenador se compone de una gran cantidad de celdillas, y cada celdilla contiene sólo una unidad de información. En este sentido es un vasto almacén de bits. Un ordenador actual de prestaciones modestas almacena y ma-

[1]Mientras esté alimentada por la corriente eléctrica.

neja varios miles de millones de bits. Por ejemplo, un CD-ROM, que es un tipo de memoria de ordenador, puede almacenar cerca de 6.000 millones de bits.

Puesto que estamos hablando de órdenes de magnitud tan elevados, es frecuente utilizar ciertos prefijos para expresar las enormes capacidades de almacenamiento con las que trabajamos en los ordenadores actuales:

Prefijo	**Número de bits**		**Aproximadamente**
*Kilo*bit (Kb.)	1.024	2^{10}	$\approx 10^3$ (mil)
*Mega*bit (Mb.)	$1.024 \times 1.024 = 1.048.576$	2^{20}	$\approx 10^6$ (millón)
*Giga*bit (Gb.)	$1.024 \times 1.024 \times 1.024 = 1.073.741.824$	2^{30}	$\approx 10^9$ (mil millones)
*Tera*bit (Tb.)	$1.024 \times 1.024 \times 1.024 \times 1.024 = 1.099.511.627.776$	2^{40}	$\approx 10^{12}$ (billón)

Cuadro 3.1. Unidades de medida expresadas en función de múltiplos de bits

3.2.2. Direcciones y agrupamientos

En principio, cada bit de la memoria es accesible mediante un número que identifica la celdilla en la que está almacenado: su *posición* o *dirección*. Por tanto, el procesador del ordenador puede hacer preguntas tales como "¿qué valor tiene ahora el bit que ocupa la posición 653.373.106?" o ejecutar acciones tales como "haz que el bit con dirección 1.015 valga 1".

Pudiendo representar únicamente un bit en cada celdilla se dispone de una capacidad de expresión reducida: raramente tiene interés manejar un bit aislado. Por esta razón, es habitual que los bits no se consideren individualmente, sino en grupos, y que la información se codifique utilizando *grupos de bits*. Los ordenadores organizan la memoria como una serie de grupos de bits *de tamaño fijo*. Mediante estas agrupaciones de bits de tamaño fijo, ya se dispone de una *gran capacidad expresiva* para representar instrucciones y datos en la memoria principal, ya que cada grupo puede contener múltiples combinaciones de ceros y unos.

Fíjate en lo siguiente. La "potencia expresiva" de un bit es muy pequeña: sólo puede representar dos valores o estados diferentes. Con un bit solamente podría codificar los elementos de un conjunto de cardinal 2. Consideremos ahora una agrupación de 2 bits. Las posibles combinaciones que puedo realizar con 2 bits son: 00, 01, 10 y 11. Es decir, si considero todas las secuencias posibles que puedo formar con 2 bits, entonces podría llegar a representar hasta 4 valores distintos. Observa que, simplemente añadiendo un bit, he duplicado la potencia expresiva.

¿Y si considero 4 bits? Entonces tendré estas posibles secuencias: 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110 y 1111. En total, 16. En general, si agrupo n bits, entonces podré representar hasta 2^n valores distintos, puesto que podré formar 2^n secuencias distintas de bits. Cuantos más bits agrupe, entonces, mi "potencia expresiva" será mayor: multiplicaré por 2 la que ya tenía antes si añado un nuevo bit a la agrupación.

Aquí radica la potencia del bit. Para que te hagas una idea: con 8 bits puedo representar 2^8 (256) valores distintos; con 16 bits puedo representar 2^{16} (65.536) valores distintos y con 32 puedo representar 2^{32} $(4.294.967.296)$ valores distintos. Los incrementos son exponenciales. Luego, en la práctica, tenemos la "potencia expresiva" su-

ficiente como para representar casi cualquier tipo de información: sonido, imágenes, vídeo…

No obstante, en informática, la agrupación de bits más frecuente es la de 8 bits. Es tan frecuente que ha sido bautizada: *cada 8 bits forman un byte* (u octeto). De hecho, los bits se agrupan en bytes y los incrementos de "potencia expresiva" se consiguen agrupando bytes. Así, un byte arbitrario podría ser, por ejemplo 01101110.

Luego, en realidad, no sucede que en la memoria principal del ordenador cada bit individual tenga una dirección, sino que es cada grupo de 8 bits o byte el que tiene una dirección en memoria. Por consiguiente, preguntas como las de antes se formulan más bien así: "¿qué valor tiene ahora el byte (los 8 bits) que ocupa la posición 653.373.106?", o si indicamos que se ejecute una acción: "haz que el byte con dirección 1.015 valga 00110101".

En realidad, cuando se quiere indicar la capacidad de memoria RAM (principal) de un ordenador, o bien la de un medio de almacenamiento secundario (por ejemplo un disco duro), se expresa en múltiplos de bytes:

Término	**Número de bits**	**Aproximadamente**
*Kilo*byte (KB)	$8 \times 1.024 = 8.192$	$\approx 10^3$ (mil) bytes
*Mega*byte (MB)	$8 \times 1.024^2 = 8.388.608$	$\approx 10^6$ (millón) bytes
*Giga*byte (GB)	$8 \times 1.024^3 = 8.589.934.592$	$\approx 10^9$ (mil millones) bytes
*Tera*byte (TB)	$8 \times 1.024^4 = 8.796.093.022.208$	$\approx 10^{12}$ (billón) bytes

Cuadro 3.2. Unidades de medida expresadas en función de múltiplos de bytes

Fíjate en las siguientes proporciones:

- Unos 1.000 bytes (en realidad, 1.024) son 1 KB.
- Unos 1.000 KB (en realidad, 1.024) son 1 MB.
- Unos 1.000 MB (en realidad, 1.024) son 1 GB.

En resumen, la respuesta a la pregunta "¿qué clase de información podemos representar con bits y grupos de bits?" es: "prácticamente cualquier tipo de información". En este capítulo nos encargaremos de ver cómo se puede representar y manipular información con bits. A lo largo del capítulo iremos viendo cómo se representan datos numéricos, caracteres, texto, imágenes, sonido, vídeo… Empezaremos por lo más básico (números) y abarcaremos hasta lo más complejo (vídeo), haciendo más hincapié en lo fundamental (números, caracteres y texto).

3.3. SISTEMAS DE REPRESENTACIÓN POSICIONAL

Suponemos que te habrás dado cuenta del hecho de que *tan sólo utilizando diez dígitos* podemos representar *un conjunto infinito de valores* como es el de los números naturales. Somos capaces de representar números arbitrariamente grandes gracias a que *la posición de los dígitos en un número cambia su "peso"* ("significado").

Ejemplo 1. El número 2.105 en realidad se interpreta así:

$$2.105 = 2 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 5 \times 10^0$$

□

Cada posición numérica presenta un *peso* (significado) asociado. La posición p (contando desde la derecha y empezando por 0) tiene peso 10^p. El valor 10 es la base del sistema de numeración decimal.

En resumen: todo número de un sistema en base b se puede expresar mediante una secuencia de símbolos, los cuales contribuyen al valor total del número con un valor parcial que depende de (1) el símbolo en sí, llamado *coeficiente*, y (2) de la posición que ocupa en la secuencia, llamada *peso*. Es decir, podemos considerar que el número en base b

$$\ldots n_3\, n_2\, n_1\, n_0\, n_{-1}\, n_{-2} \ldots$$

está representando el valor

$$\ldots + n_3 \cdot b^3 + n_2 \cdot b^2 + n_1 \cdot b^1 + n_0 \cdot b^0 + n_{-1} \cdot b^{-1} + n_{-2} \cdot b^{-2} + \ldots$$

Ejemplo 2.

Sistema decimal:

$$7.849{,}36 = 7 \cdot 10^3 + 8 \cdot 10^2 + 4 \cdot 10^1 + 9 \cdot 10^0 + 3 \cdot 10^{-1} + 6 \cdot 10^{-2}$$

Sistema binario:

$$101{,}01 = 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 + 0 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

□

Has de tener en cuenta que no todos los sistemas de numeración son posicionales. El sistema romano de numeración, por ejemplo, no lo es: el número 2.105 se escribe como MMCV en el sistema numérico romano. Con sistemas no basados en la posición como éste se complica mucho el poder representar números arbitrariamente grandes: puesto que el sistema romano de numeración *no* es posicional, se requieren más símbolos para representar números cada vez más grandes. Así, por ejemplo, para representar números hasta 100 se utilizan los símbolos {I,V,X,L,C}; pero para representar números hasta 1.000 se requieren dos símbolos más: D y M.

Es posible definir sistemas de numeración con otras bases aparte de la base 10, empleada en el sistema decimal. Uno especialmente interesante en informática[2] es el *sistema binario*, es decir, el sistema que utiliza la *base 2*. En base 2, cada dígito (símbolo) puede ser un 0 o un 1, es decir, cada dígito (símbolo) es un bit.

[2]Recuerda que los ordenadores son sistemas digitales binarios.

3.3.1. El sistema binario

Un sistema de numeración en base b utiliza b símbolos para representar los números. Así, por ejemplo, sabemos que el sistema decimal ($b = 10$) utiliza el conjunto de los dígitos { `0, 1, 2, 3, 4, 5, 6, 7, 8, 9` }. El sistema binario ($b = 2$) utiliza el conjunto { `0, 1` }. En el caso del sistema binario, la posición p (recuerda; contando desde la derecha y empezando en 0) tiene peso 2^p.

Ejemplo 3. El número binario 1001 representa el valor decimal 9:

$$1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 9$$

□

Un par de cuestiones importantes de nomenclatura ("jerga informática"): el bit que está más a la izquierda recibe el nombre de *bit más significativo* ya que es el que tiene mayor peso, mientras que al que está más a la derecha se le llama *bit menos significativo* (es el que tiene menos peso).

bit más significativo ↓   bit menos significativo ↓

1 0 1 1 1

En general, en el sistema binario se requerirán más cifras que en el sistema decimal para representar un mismo valor. En la siguiente tabla se muestran algunas equivalencias entre números en binario y en decimal.

binario	**decimal**	**binario**	**decimal**	**binario**	**decimal**	**binario**	**decimal**
0	0	1010	10	10100	20	11110	30
1	1	1011	11	10101	21	101000	40
10	2	1100	12	10110	22	110010	50
11	3	1101	13	10111	23	111100	60
100	4	1110	14	11000	24	1000110	70
101	5	1111	15	11001	25	1010000	80
110	6	10000	16	11010	26	1011010	90
111	7	10001	17	11011	27	1100100	100
1000	8	10010	18	11100	28		
1001	9	10011	19	11101	29		

Finalmente, convendría aclarar ciertas cuestiones de notación. Observa que ciertos números son "ambiguos". Fíjate en que tanto el 1 como el 0 son símbolos (dígitos) que se emplean tanto en el sistema decimal como en el binario. Si un número concreto sólo consta de unos y ceros, no está claro qué número queremos representar (dependerá del valor de la base):

- ¿10 es diez (interpretación en base 10) o dos (interpretación en base 2)?

- ¿111 es ciento once (interpretación en base 10) o siete (interpretación en base 2)?

Para dejar claro lo que estamos representando en estos casos, utilizaremos una notación en la que, cuando convenga, emplearemos un subíndice para indicar la base:

- 10_2 es dos, pues estamos indicando con el subíndice que la base es 2 (binario).
- 10_{10} es diez, pues estamos indicando con el subíndice que la base es 10 (decimal).

Ejercicio 1. ¿Qué valores decimales representan los siguientes números binarios?

a) 1111_2

b) 1_2

c) 0_2

d) 0111_2

e) 10_2

f) 1000_2

Ejercicio 2. Responde a estas cuestiones:

a) ¿Cuál es el valor más grande que podemos expresar con cuatro bits?,

b) ¿y con ocho?,

c) ¿y con n, siendo n un entero positivo?

Ahora, lo que nos interesa conocer del sistema binario son las operaciones aritméticas y lógicas básicas que podemos realizar y las transformaciones desde el sistema decimal al binario.

3.3.2. Conversión de números decimales a otra base

Debes saber que, en general, podemos transformar un número decimal entero a otra base *dividiendo reiteradamente por la base en cuestión*. Así, la transformación *de decimal a binario* la haremos de la siguiente forma:

1.- Realizar una división entera entre `2` del número decimal entero. Es decir, dividir entre `2` el número decimal entero dejando el resultado como cociente y resto. El resto siempre será `0` ó `1`, ya que el divisor es `2`.

2.- Considerar el cociente obtenido como nuevo número en decimal a dividir y aplicar el paso 1.

3.- Repetir los pasos 1 y 2 hasta que el cociente obtenido sea 1. Siempre que se sigan estos pasos se llegará a obtener un cociente 1, excepto si el número entero en decimal a dividir es 0 ó 1 (en cuyo caso su representación en binario es la misma que en decimal).

4.- La representación del número entero decimal en binario se obtiene como *la secuencia inversa de los restos obtenidos*, siendo el último cociente la cifra más significativa (más a la izquierda) del número en binario.

Ejemplo 4. Convirtiendo el número entero decimal 108 a base 2:

108	2					
08	54	2				
0	14	27	2			
	0	07	13	2		
		1	**1**	6	2	
				0	3	2
					1	**1**

Solución: 108_{10} es 1101100_2. □

Ejercicio 3. Convierte los siguientes números decimales a base 2:

a) 0
b) 1
c) 18
d) 128
e) 127
f) 863
g) 2^{10}

3.3.3. Suma binaria

Un bit puede representar dos cosas distintas: 1) una cifra de un número en el sistema binario; 2) una variable lógica.

Es decir, un bit sufrirá tratamientos claramente diferenciados dependiendo de si la operación en la que participa es aritmética o lógica. La diferencia más clara consiste en que en una operación aritmética *la aplicación del operador sobre los bits de una determinada posición puede depender del resultado de la aplicación del operador en los bits precedentes*, mientras que *en una operación lógica* la aplicación del operador sobre los bits de una determinada posición *sólo depende de ellos*.

Queremos aprender a realizar operaciones aritméticas en binario. En particular, aunque pondremos ejemplos de las otras operaciones básicas, nos centraremos en la suma. No obstante, antes de aprender a operar en base 2, hemos de conocer las tablas básicas de operaciones bit a bit. Las operaciones aritméticas en binario son análogas a las realizadas en decimal:

suma
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0 (y acarreo 1)

resta
0 − 0 = 0
0 − 1 = 1 (y sumamos 1 al sustraendo)
1 − 0 = 1
1 − 1 = 0

producto
0 × 0 = 0
0 × 1 = 0
1 × 0 = 0
1 × 1 = 1

división
0 : 0 no existe
0 : 1 = 0
1 : 0 no existe
1 : 1 = 1

Ahora ya sabemos realizar las operaciones básicas en base dos, en particular podemos sumar números de modo similar a como lo hacemos en base diez: llevando el exceso (acarreo) de una columna a la siguiente. Ya sabes: aquello de "…ocho y seis, catorce y llevo una…" ahora se convierte en "…uno y uno, cero y llevo uno…".

Ejemplo 5. Veamos un ejemplo de suma:

		1	1		Acarreo
	1	0	1	1	
+			1	1	
	1	1	1	0	

□

Como hemos dicho, análogamente a las operaciones en decimal, el acarreo en la suma significa sumar dicho acarreo al resultado de la suma de las siguientes cifras. Sin embargo, en la resta implica sumar el acarreo a las correspondientes siguientes cifras del sustraendo (o equivalentemente, restar el acarreo a las correspondientes siguientes cifras del minuendo).

El método de resta de dos números de varias cifras requiere que el minuendo sea siempre el número cuyo valor absoluto es mayor y correspondientemente que el sustraendo sea el número con menor valor absoluto. Manipulando adecuadamente los signos de los números (minuendo, sustraendo y resultado) se puede realizar la resta de cualquier par de números.

Ejemplo 6. Veamos ejemplos de operaciones aritméticas con números binarios.

```
 1 1 1
   1 0 1 1          1 0 1 1 0 1
 + 1 1 1 0          1     1 1
 ---------        -   1 0 1 1 1
 1 1 0 0 1        -------------
                    0 1 0 1 1 0

            1 0 1 1 0
       ×        1 0 1
    -----------------
            1 0 1 1 0
          1
          0 0 0 0 0
    + 1 0 1 1 0
    -----------------
      1 1 0 1 1 1 0
```

Un 1 pequeño sobre una cifra representa el acarreo de la cifra anterior.

```
  1 1 1 0 1 , 0 1 | 1 1
- 1 1             ----------
-----               1 0 0 1 , 1 1
  0 0 1 0 1
      1
  -     1 1
  ---------
      0 1 0 0
        1 1
      -   1 1
      -------
        0 0 1 1
        -   1 1
        -------
            0 0
```

□

Ejercicio 4. Calcula las siguientes sumas de números binarios:

a) $101011_2 + 100101_2$

b) $1_2 + 1001_2$

c) $10000_2 + 10000_2$

d) $101010_2 + 10101_2$

e) $101010_2 + 101010_2$

3.3.4. Circuitería para las sumas

Es posible implementar circuitos capaces de llevar a cabo *operaciones lógicas* con bits: *1 es verdadero*, *0 es falso*. Recuerda que un bit puede representar una variable lógica. Recuerda, igualmente, que, a diferencia de las operaciones aritméticas, la aplicación del operador sobre los bits de una determinada posición sólo depende de ellos.

Hay tres operaciones lógicas básicas: la *conjunción* o *producto lógico* (∧, **Y** o **AND**), la *disyunción* o *suma lógica* (que puede representarse como ∨, **O** u **OR**) y la *complementación* o *negación* (¬, **NO** o **NOT**). También es frecuente utilizar la negación de la disyunción (**NOR**), la negación de la conjunción (**NAND**) y la *disyunción lógica exclusiva* u *o-exclusivo* (**XOR**). A las operaciones lógicas también se las denomina *operaciones booleanas* (por el *álgebra de Boole*). A continuación, se muestran las tablas de verdad de las operaciones básicas, además de la del o-exclusivo (por su relevancia).

A	B	A **OR** B (A ∨ B)	A **AND** B (A ∧ B)	**NOT** A ¬A	A **XOR** B (A ∧ ¬B) ∨ (¬A ∧ B)
0	0	0	0	1	0
0	1	1	0	1	1
1	0	1	0	0	1
1	1	1	1	0	0

Mediante la tabla de verdad de cada operador conocemos el resultado que se obtiene al aplicar el operador a dos operandos que son bits individuales. Para aplicar el operador a dos secuencias de bits (siempre de la *misma longitud*), se emparejarán los bits que ocupan la misma posición en cada secuencia y se obtendrá el valor de la tabla de verdad del operador correspondiente a dicho par de bits.

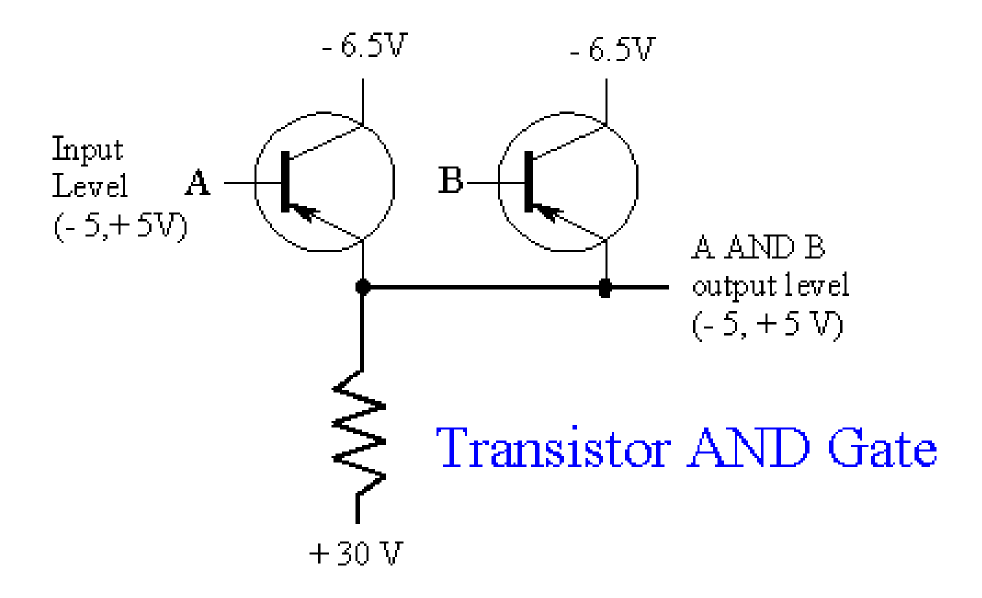


Figura 3.2. Sencillo circuito que implementa una puerta lógica AND con transistores

Ejemplo 7. Veamos ejemplos de estas operaciones lógicas con secuencias de bits.

	1	0	1	1	0	1	0	1	1
OR	1	1	1	0	0	1	1	0	1
	1	1	1	1	0	1	1	1	1

	1	0	1	1	0	1	0	1	1
AND	1	1	1	0	0	1	1	0	1
	1	0	1	0	0	1	0	0	1

NOT	1	0	1	1	0	1	0	1	1
	0	1	0	0	1	0	1	0	0

	1	0	1	1	0	1	0	1	1
XOR	1	1	1	0	0	1	1	0	1
	0	1	0	1	0	0	1	1	0

□

¿Por qué son importantes las operaciones lógicas? Pues porque, en la práctica, cada operación lógica puede implementarse empleando circuitería electrónica sencilla: bastan un par de transistores y una resistencia, como puedes observar en la Figura 3.2.

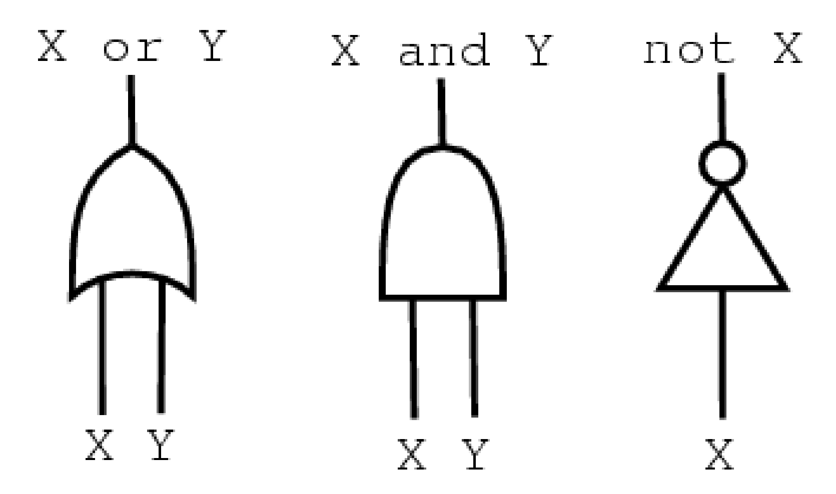
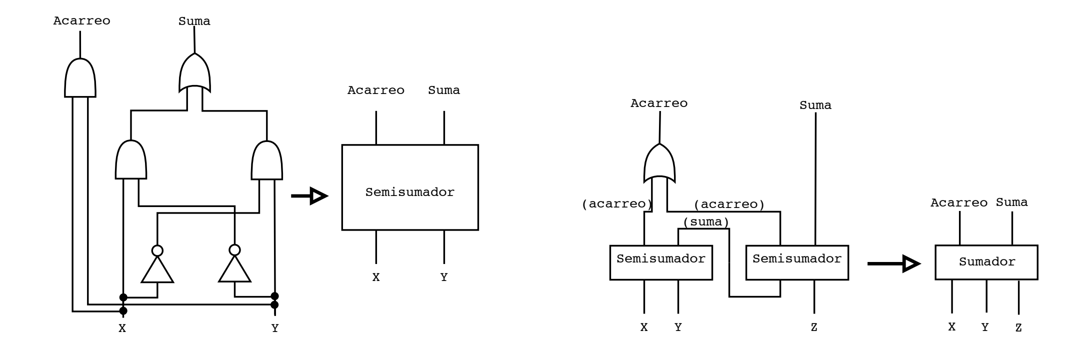


Figura 3.3. Representación de los circuitos que implementan las operaciones lógicas básicas: OR, AND y NOT

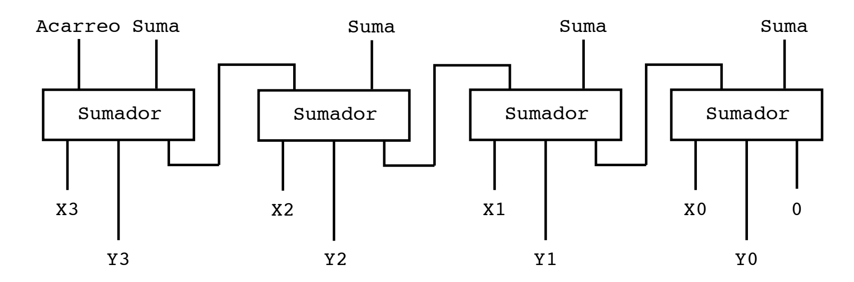
Para que te hagas una idea de la enorme cantidad de sencillos circuitos capaces de implementar rápidas y eficientes puertas lógicas que hay en los circuitos impresos de los ordenadores, baste decir que en un chip caben millones de transistores. En dichos circuitos electrónicos lógicos (es decir, que manejan bits) o circuitos impresos, las operaciones lógicas se representan gráficamente con unos símbolos universalmente utilizados. La Figura 3.3 muestra estos símbolos.

Debes tener presente que la potencia de un ordenador reside en la posibilidad de implementar operaciones más complicadas que las denominadas básicas, mediante la



(a) Un circuito semisumador (suma 2 bits) construido con puertas lógicas

(b) Un circuito sumador (suma 3 bits) construido con dos semisumadores y una puerta lógica OR



(c) Un circuito capaz de sumar dos números de 4 bits construido con sumadores

Figura 3.4. Circuitería básica para efectuar sumas

combinación de varios circuitos lógicos. Por ejemplo, la circuitería con la que se pueden implementar circuitos capaces de realizar sumas de dos operandos (secuencias de bits) de anchura fija es relativamente sencilla. La Figura 3.4 ilustra este proceso, tomando como objetivo la realización de un circuito capaz de sumar correctamente dos secuencias compuestas de cuatro bits.

Como vimos anteriormente, mediante la presentación de una serie de ejemplos, podemos efectuar otras operaciones aritméticas con números binarios, extendiendo de forma natural los métodos que aplicamos al operar con números en base 10.

Ejercicio 5. Efectúa las siguientes operaciones con números binarios:

a) $10011_2 - 101_2$

b) $1001_2 \times 1001_2$

c) $110000_2/10_2$

Hemos expuesto el procedimiento que hay que llevar a cabo para realizar las operaciones aritméticas básicas en binario: suma, resta, multiplicación y división. No obstante, nos centraremos fundamentalmente en la suma para nuestros propósitos ilustrativos. Tengamos en cuenta que, por ejemplo, y para simplificar la circuitería de los ordenadores, *la resta se plantea como una suma en la que el segundo sumando cambia*

de signo. Esto nos conduce al siguiente planteamiento: *¿cómo podemos representar números con signo?*

3.3.5. Números con signo

De momento, todos los números que hemos representado utilizando el sistema binario (base 2) son positivos. Para representar los números negativos, podríamos emplear un procedimiento similar al que adoptamos cuando trabajamos con el sistema decimal y al cual estamos habituados: introducir el símbolo "$-$" (signo negativo).

Sin embargo, esta solución no es viable. En un sistema digital binario como es el ordenador, surge un problema: sólo podemos representar y almacenar dos símbolos, el 1 y el 0, con lo cual no tenemos capacidad para representar el signo, ya que el signo es un símbolo adicional.

¿Cuál es la solución a este problema? La solución consiste en representar el signo mediante *un bit* (como siempre) *en una posición determinada*: el valor 0 en dicha posición indica *positivo*, mientras que el valor 1 en dicha posición indica *negativo*. Y ¿cuál es la posición elegida? No hay una establecida de antemano, ya que depende de varias circunstancias (la arquitectura o el tipo de datos con el que estemos trabajando, por ejemplo). Pero la idea básica consiste en trabajar con *números de ancho fijo* y codificar el signo *en el bit más significativo*.

Como se ha comentado, no hay un ancho estándar establecido universalmente, pero típicamente se emplean 8, 16, 32 ó 64 bits[3]. Realmente, la idea de fijar la anchura no es un artificio forzado, ya que la tecnología empleada en los ordenadores actuales (arquitectura) hace natural el agrupamiento de bits en bloques de este tamaño (8, 16, 32 ó 64 unidades). Un ejemplo: si empleamos una anchura fija de 8 bits, entonces el valor 2 se codificaría realmente como (rellenando con ceros a la izquierda, es decir no significativos):

0	0	0	0	0	0	1	0

Parece que hemos encontrado la solución a nuestro problema de la representación del signo. No obstante, habrás de tener en cuenta que fijar una anchura determinada para los números presenta ciertos problemas relacionados con el *desbordamiento* (el hecho de superar el número de bits máximo que viene determinado por la anchura para representar un número) al realizar operaciones aritméticas. Un par de ejemplos:

- La suma de dos números de p bits puede ocupar $p + 1$ bits. Se dice entonces que se ha producido un *desbordamiento* siempre que se produce un acarreo al final, es decir, al sumar los dos bits más significativos.

  Esta situación está prevista por la propia circuitería, ya que la unidad aritmética del ordenador da un aviso cuando se produce un desbordamiento, de modo que el programador[4] pueda tomar la decisión más acertada.

[3] Estas anchuras se consideran estándar.

[4] Estamos hablando de programación a "bajo nivel", ya que los lenguajes de programación de alto nivel "enmascaran" estas situaciones: el programador debe controlar los valores con los que opera en función del tipo de datos empleado en la operación.

- El producto de dos números de p bits puede ocupar $2 \times p$ bits.

  Las operaciones de multiplicación devuelven un número codificado en el doble de bits[5].

Existen varias codificaciones distintas para representar números con signo, empleando una anchura fija. Nosotros estudiaremos brevemente las representaciones en *complemento a 1* y *complemento a 2* por su sencillez y practicidad y porque, de hecho, la representación mediante complemento a 2 es la que se emplea realmente en los ordenadores.

Complemento a 1

En la notación en complemento a 1 representamos un valor decimal dado así:

- Se escribe, en primer lugar, el *valor absoluto* del número decimal en binario (se calcula mediante divisiones sucesivas por 2, tal como hemos visto).
- A continuación, si el número decimal dado es negativo, *se invierten los bits* (los unos pasan a ser ceros y viceversa).

Por interés didáctico, asumiremos, en adelante, que trabajamos con una anchura fija de 4 bits. Veamos un par de ejemplos empleando esta anchura: el valor 2 se codifica como 0010 (no hace falta invertir los bits; es positivo) mientras que el valor -2 se codificaría como 1101 (invertimos los bits ya que es negativo).

Ejercicio 6. Convierte a base 10 estos números binarios en complemento a uno de cuatro bits:

a) 1101
b) 0101
c) 0000
d) 1111

Ejercicio 7. Responde a esta pregunta:

¿Cuáles son los números mayor y menor que podemos expresar en complemento a uno de cuatro bits?

Empleando la representación de signo y magnitud la suma no funciona correctamente, problema que no aparece si utilizamos la representación en complemento a 1. Veamos un ejemplo de suma empleando la representación en complemento a 1: 1_{10} más -1_{10} da 0:

[5] Tenga en cuenta el lector lo expresado en la nota al pie anterior a ésta.

$$\begin{array}{rcccc} & 0 & 0 & 0 & 1 \\ + & 1 & 1 & 1 & 0 \\ \hline & 1 & 1 & 1 & 1 \end{array}$$

mientras que con signo y magnitud tenemos que: $0001 + 1001 = 1010$, lo cual no es correcto. Desgraciadamente, la suma no siempre funciona bien. Veamos lo que ocurre si sumamos 3 y -2. El resultado debería ser 1.

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 1 & 1 & 0 & 1 \\ \hline (1) & 0 & 0 & 0 & 0 \end{array}$$

¿Qué ha ocurrido? ¿Por qué el resultado es 0 y no 1 como debiera? Observa que aparece un bit adicional porque hay un *desbordamiento* en la suma. *Cuando hay un desbordamiento, el resultado es incorrecto* pero se soluciona fácilmente: *basta con sumar un uno al resultado*. Así, el resultado correcto del ejemplo es 0001.

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 1 & 1 & 0 & 1 \\ \hline (1) & 0 & 0 & 0 & 0 \\ & & \downarrow & & \\ & 0 & 0 & 0 & 0 \\ + & 0 & 0 & 0 & 1 \\ \hline & 0 & 0 & 0 & 1 \end{array}$$

Ejercicio 8. Efectúa las siguientes operaciones codificando los números en base 10 a sus equivalentes en complemento a uno con cuatro bits:

a) $3 + 2$

b) $3 - 2$

c) $1 + 1$

d) $1 - 1$

e) $7 - 7$

f) $-2 - 2$

g) $-2 + 3$

Esperamos que algunos de los ejemplos y ejercicios planteados habrán servido para esbozar cuáles son los problemas que aparecen con el empleo de la representación en complemento a 1:

(a) Hay dos formas diferentes de representar el cero: 0000 y 1111. Ello obliga a tener esta situación en cuenta.

(b) La suma es relativamente lenta: una vez realizada debe comprobarse el valor del acarreo y, si es 1, sumárselo al resultado (por consiguiente, realizando dos sumas en la práctica).

Complemento a 2

Precisamente para solucionar estos problemas surgió *el complemento a 2*. La notación en complemento a 2 permite que las sumas sean más sencillas y rápidas y presenta una forma única para el cero. En conclusión: *es la que se usa realmente en los ordenadores*. La representación en complemento a 2 consiste en hacer lo mismo que en complemento a 1 salvo que, cuando el número a representar es negativo (cambio de signo del número), hacemos lo siguiente:

1.- Complementamos (invertimos) todos los bits del número (los unos pasan a ser ceros y viceversa).

2.- Y, finalmente, sumamos 1 al resultado.

Veamos un ejemplo. Para representar el valor -4 (con 4 bits) partimos de la representación binaria del número 4: 0100; se complementan sus bits: 1011; y se suma 1 al resultado: 1100. De esta manera, la suma en complemento a 2 se convierte en *una suma convencional* en la que *descartamos* los bits de exceso (si los hay), es decir descartamos el acarreo en la suma de los bits más significativos si lo hubiere. Para demostrarlo, sumemos 3 y -2. El resultado debe ser 1:

$$\begin{array}{rcccc} & 0 & 0 & 1 & 1 \\ + & 1 & 1 & 1 & 0 \\ \hline (1) & 0 & 0 & 0 & 1 \end{array}$$

Ejercicio 9. Efectúa las siguientes operaciones en complemento a 2 con 4 bits:

a) $3 + 2$

b) $3 - 2$

c) $1 + 1$

d) $1 - 1$

e) $7 - 7$

f) $-2 - 2$

g) $-2 + 3$

3.3.6. Números con parte fraccionaria

Los números fraccionarios, también conocidos como números con decimales, son, en realidad, *los números reales*. Éstos se representan en los sistemas numéricos posicionales mediante la utilización de pesos negativos para la parte decimal. Veamos un par de ejemplos, uno en decimal y otro en binario.

Ejemplo 8. En base 10, el número $13{,}14$ es

$$1 \times 10^1 + 3 \times 10^0 + 1 \times 10^{-1} + 4 \times 10^{-2}$$

□

Ejemplo 9. En base 2, el número $10{,}01$ es

$$1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}$$

es decir, es el número $2{,}25$ en base 10. □

3.3.7. Conversión de números fraccionarios en base 10 a base 2

La transformación de decimal a binario la haremos de la siguiente forma. Supongamos que tenemos un número cualquiera en decimal con parte entera y parte fraccionaria. Obtendremos la parte entera del número en binario con la parte entera del número en decimal, y su parte fraccionaria con la parte fraccionaria del número en decimal. Procederemos separadamente como sigue.

1.- Obtención de la parte entera del número en binario mediante divisiones sucesivas por 2 (tal como hemos visto).

2.- Obtención de la parte fraccionaria del número en binario así:

(a) Multiplicar por 2 la parte fraccionaria (decimal) del número. Así, la parte entera del resultado que se obtenga siempre será 0 ó 1.

(b) Tomar la parte fraccionaria (decimal) del resultado (o sea, no teniendo en cuenta el dígito entero obtenido) y aplicar el paso (a).

(c) Repetir los pasos (a) y (b) hasta que la parte fraccionaria (decimal) del resultado sea 0 ó hasta que identifiquemos que una subsecuencia de partes enteras que se van obteniendo es repetición de otra subsecuencia anterior. En este último caso, lo que obtendremos es un número fraccionario en binario con período. Hay que tener en cuenta, que *una parte fraccionaria que en decimal se expresa con una secuencia finita de cifras puede dar lugar a una secuencia infinita (periódica) de cifras en binario*.

(d) La parte fraccionaria (decimal) en binario se obtiene como la secuencia directa de las partes enteras obtenidas en los sucesivos productos.

Ejemplo 10. Conversión del número decimal $4{,}125$ a binario:

- la parte entera es 100 (se puede calcular por divisiones sucesivas),
- y la parte decimal se calcula así:
  1. $0{,}125 \times 2 = \mathbf{0}{,}25$, así que 0 es el primer decimal.
  2. $0{,}25 \times 2 = \mathbf{0}{,}5$, así que 0 es el segundo decimal.
  3. $0{,}5 \times 2 = \mathbf{1}{,}0$, así que 1 es el tercer decimal,

- y como la parte decimal es nula (1,**0**), hemos acabado y el número resultante es 100,001.

□

Ejemplo 11. Transformación a binario del número decimal 57,0625.

Consideramos primero la parte entera del número: 57.

```
57 | 2
17   28 | 2
 1   08   14 | 2
      0    0   7 | 2
               1   3 | 2
                   1   1
```

Por tanto, la parte entera del número en binario será: 111001.

Ahora transformamos a binario su parte fraccionaria: 0,0625.

```
 0,0625        0,125       0,25       0,5
    × 2          × 2        × 2       × 2
 ------        -----       ----       ---
 0,1250        0,250       0,50       1,0
```

Con ello, obtenemos la parte fraccionaria: 0,0001.

Y finalmente, el número en binario correspondiente al número decimal dado es 111001,0001. □

Ejercicio 10. Codifica en binario los siguientes números en base 10:

- 10,25
- 128,125
- 1,75
- 2,875
- 3,3

3.3.8. Notación exponencial o de coma flotante

La unidad de trabajo y de intercambio de información entre la CPU y la memoria principal es la *palabra* (secuencia de bits de anchura fija). La longitud de palabra del procesador y de la memoria depende de la estructura física del ordenador. No obstante, conviene que esta longitud sea un múltiplo entero del número de bits utilizado para

representar un carácter (8)[6]. De esta forma se puede aprovechar mejor la ocupación de la memoria.

La representación interna de los datos, además de la estructura física del ordenador, depende del lenguaje de programación y del compilador/intérprete utilizados. Tras abordar la representación de números enteros (con o sin signo) y estudiar la forma en que los números decimales pueden codificarse en el sistema binario, vamos a abordar una representación eficiente y compacta de los números reales, bueno, *en realidad de la aproximación que a ellos realizan los ordenadores y que denominamos números de coma (punto) flotante o "flotantes" a secas*.

Los datos correspondientes a números de coma flotante, se representan basándose en la *notación exponencial o científica*; es decir, con el siguiente formato:

- signo
- mantisa
- base
- exponente

Estas partes se combinan así:

$$\text{signo} \quad \text{mantisa} \times \text{base}^{\text{exponente}}$$

Por ejemplo, la transformación inmediata del número real en notación fraccionaria $347,03$ a su notación exponencial es $+347,03 \times 10^0$. Sin embargo, existen varias representaciones posibles para un mismo número en coma flotante. Para nuestros propósitos docentes, supongamos que estamos trabajando con el sistema decimal.

Ejemplo 12. Consideremos todas las representaciones en notación exponencial del número $347,03$. Serán las siguientes:

$$\ldots = +3470,3 \times 10^{-1} = +347,03 \times 10^{0} = +34,703 \times 10^{1} = +3,4703 \times 10^{2} =$$
$$+0,34703 \times 10^{3} = +0,034703 \times 10^{4} = +0,0034703 \times 10^{5} = \ldots$$

□

Ejemplo 13. Éstas son algunas de las formas en que podemos representar el número -50:

[6]Hoy en día, lo más común en el mundo PC es usar una longitud de 32 bits, aunque ya empiezan a aparecer los primeros procesadores de 64 bits.

signo y mantisa	**base**^{exponente}
-50000	10^{-3}
-5000	10^{-2}
-500	10^{-1}
-50	10^{0}
$-5{,}0$	10^{1}
$-0{,}5$	10^{2}
$-0{,}05$	10^{3}

□

Precisamente, para evitar la ambigüedad y favorecer la eficiencia, se establecen los siguientes criterios, que restringen aún más el formato de representación:

(a) Normalización de la mantisa. El exponente se ajusta para que la mantisa sea un número menor que 10 (*en general, que la base* del sistema de numeración con el que estamos trabajando), y para que *su parte entera no sea nula*.

Por ejemplo, la *única* representación del número $347{,}03$ que tiene la mantisa normalizada es $3{,}4703 \times 10^2$ ya que es la única que cumple ser menor que la base, 10, ($3{,}4703 < 10$) y, al mismo tiempo, cumple que su parte entera no sea 0 ($0,34703 \times 10^3$ no cumpliría esto último, por ejemplo). Siguiendo con el otro ejemplo, la forma normalizada para -50 es $-5{,}0 \times 10^1$.

(b) La base de esta representación exponencial siempre está predeterminada, con lo que no será necesario almacenarla explícitamente para representar el dato real. Obviamente, si trabajamos en el sistema binario, la base siempre será 2.

A partir de estas consideraciones, una palabra de memoria se fragmenta de la siguiente forma para representar el signo, la mantisa y el exponente de un número real:

signo	*exponente*	*mantisa*

El signo, al igual que ocurría en la representación de los enteros con signo, sólo ocupa un bit, y su valor será `0` para los números positivos y `1` para los negativos. La mantisa se almacena en binario natural y está normalizada. Así, no será necesario almacenar la parte entera (que siempre es `1`) ni la coma, siendo el bit más a la izquierda de la mantisa la cifra fraccionaria más significativa no nula.

Los bits dedicados al exponente, n_e, representarán un número entero en el rango

$$\{-2^{n_e-1}-1,\ldots,-1,0,1,\ldots,+2^{n_e-1}-1\}.$$

No obstante, como en este rango aparecen números negativos, hay que especificar una forma de almacenar que tenga en cuenta el signo del exponente. Una posibilidad consistiría en almacenar el rango de números correspondiente al exponente como los enteros con signo (un bit para el signo y el resto para el valor absoluto). Pero el exponente no se almacena así, sino que se almacena en exceso a $2^{n_e-1}-1$. Dado un exponente cualquiera, e, lo que se almacena en los bits correspondientes al exponente

es $e+2^{n_e-1}-1$. De esta forma, el rango de valores almacenados es un "desplazamiento" del rango de valores del exponente que se desea representar. Es decir, el rango de valores que se almacena es $\{0, 1, \dots, 2^{n_e}-1\}$. Después, cuando se desea utilizar el valor auténtico del exponente se le resta $2^{n_e-1}-1$ al valor almacenado.

La IEEE (Institute of Electrical and Electronics Engineers) ha producido un estándar para coma flotante en binario. Es el formato **IEEE 754** que define dos tipos de números en coma flotante:

- números en coma flotante de *32 bits*,
- números en coma flotante de *64 bits*.

Aquí detallamos sólo el formato de 32 bits:

- Es un formato normalizado.
- Emplea 1 bit para el signo (el bit más significativo),
- Utiliza 8 bits para el exponente, el cual se representa como número entero con desplazamiento 127 ($2^{8-1}-1$); es decir, el 0 se escribe como 01111111 (o sea $127+0=127$); el 1 como 10000000 ($127+1=128$). Como el exponente ocupa 8 bits, el rango de valores que puede representar es:

  $$\{0,1,2,...,2^8-1\} = \{-127,...,-1,0,1,...,128\}$$

- Usa 23 bits para la mantisa (en realidad, 24, pero como el bit más significativo siempre es 1 –en el formato normalizado la parte entera no puede ser 0, por tanto en el sistema binario sólo puede ser 1– se omite).
- Y, por supuesto, la base tiene valor 2.

Ejemplo 14. El número binario $-1{,}11_2$ se representa así en coma flotante normalizada:

$$-1{,}11_2 \times 2^0$$

en el formato IEEE 754:

1	01111111	11000000000000000000000

El número $0,0000001101011_2$ se representa así en coma flotante normalizada:

$$1{,}101011_2 \times 2^{-7}$$

en el formato IEEE 754:

0	01111000	10101100000000000000000

□

Llegado este punto, el lector curioso se preguntará cuáles son las ventajas que proporciona la utilización del formato de coma flotante de la IEEE. Destaquemos algunas:

- Cubre un gran rango de valores.
  Por ejemplo, la norma IEEE 754 para coma flotante de 64 bits puede representar números entre -10^{308} y 10^{308}.
- Su precisión se "adapta" a la magnitud del número.
  Por ejemplo, en el mismo formato, podemos representar números tan pequeños como 10^{-323}.
- Es una representación estándar, con lo que podemos intercambiar informaciones de este tipo entre diferentes ordenadores.

Para concluir con este apartado, quisiéramos resaltar algunos de los conceptos importantes. Recuerda que los números de punto (o coma) flotante son, en realidad, *una aproximación* a los números reales "de verdad". Aunque grande, la capacidad de representación/almacenamiento de información en los ordenadores es *finita* y, para que te hagas una idea, sólo en el intervalo abierto $]0, 1[$ de la recta real existen infinitos números reales. De la misma manera, el sistema binario nos da una precisión determinada: recuerda el ejercicio en que un número decimal exacto como $3{,}3$ requiere una representación en binario con *infinitos decimales* (número periódico); es decir, *no es posible representarlo con exactitud*. En suma, has de ser consciente de que:

- No todo número real puede representarse como un número en coma flotante ($3{,}3$, por ejemplo):
  - Hay un rango acotado,
  - y dentro de él, errores de redondeo (falta de precisión).
- Las operaciones matemáticas afectan a la precisión de los resultados.
- Las operaciones aritméticas son costosas.
  De hecho, hasta la aparición del 486 era necesario instalar un coprocesador matemático en los ordenadores para poder efectuar operaciones en coma flotante a una velocidad razonable.

3.4. CARACTERES Y TEXTO

Ya hemos visto que es posible representar tanto números enteros como (una aproximación a los) reales usando sólo bits:

- ► En el caso de los números enteros, siempre en un rango acotado. Por ejemplo, si usamos 32 bits y consideramos emplear un bit para el signo, el número más grande que podemos representar será $2^{31} - 1 = 2.147.483.647$, aunque, eso sí, será una representación fiel (no habrá valores que no podamos representar exactamente en dicho rango).

- ► Si consideramos los números reales, entonces no sólo nos tendremos que conformar con representarlos en un rango acotado, sino también con representarlos con una precisión (número de decimales) limitada. Por ejemplo, hay infinitos números reales en $]0, 1[$, de ahí que la representación no pueda ser fiel (hay números

en el rango que no podremos representar con exactitud, por ejemplo 3,3) y hablemos de números de coma flotante (y no reales).

Siguiendo nuestra ascensión “peldaño a peldaño”, ahora toca preguntarse: ¿cómo podríamos representar las letras y los caracteres especiales (letras acentuadas, ñ, ç, etc.) mediante secuencias de bits?

3.4.1. ASCII

La idea más sencilla consiste en determinar una codificación “a priori” y almacenarla en una tabla. La primera que surgió (y que, de alguna forma, continúa vigente en la actualidad) fue la codificación *ASCII* (*American Standard for Coded Information Interchange*). Esta tabla de códigos representa, mediante secuencias de 7 bits, cada uno de 128 símbolos “internacionales” (teniendo en cuenta el alfabeto del inglés americano). El Cuadro 3.3 muestra la serie completa de códigos ASCII.

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	TAB	LF	VT	FF	CR	SO	SI
001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
010		!	”	#	$	%	&	’	(	)	*	+	,	-	.	/
011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
101	P	Q	R	S	T	U	V	W	X	Y	Z	[	/	]	ˆ	_
110	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	˜	DEL

Cuadro 3.3. La tabla de códigos ASCII que asocia secuencias de 7 bits para representar 128 símbolos distintos

Las características más destacables de esta tabla de códigos son las siguientes:

- Ciertos símbolos son caracteres de control (en la tabla se muestran con códigos de tres letras): sirven para controlar los dispositivos en los que se muestra el texto (impresoras, pantallas).

  Por ejemplo, DEL significa “borrar” (*delete*), CR es “retorno de carro” (*carriage return*), etc.

- Se respeta el *orden alfabético* tanto en las minúsculas como en las mayúsculas y en los dígitos. Esto quiere decir que si un carácter va alfabéticamente delante de otro, su código ASCII es menor que el de ese otro.

  Esta característica es muy interesante cuando se requiere ordenar datos textuales alfabéticamente (por ejemplo, la lista de alumnos matriculados por apellido): para realizarlo basta con comparar numéricamente los códigos ASCII asignados a cada letra que conforma los textos a ordenar. Es decir, “Amengual” va antes que “Morales” porque el código ASCII asignado a la “A” es numéricamente menor que el asignado a la “M”. Si tenemos “Amengual” y “Andrés”, entonces deberemos recurrir a las segundas letras para dirimir el orden: “Amengual” va antes porque el código ASCII de la “m” es menor que el de la “n”.

- Las *mayúsculas* tienen códigos que van del 1000001 al 1011010 y las *minúsculas* del 1100001 al 1111010: *se diferencian en un bit*. Esto quiere decir que es muy fácil convertir un texto escrito en mayúsculas a minúsculas y viceversa: para lo primero basta con *sumar* una cantidad fija al código binario que representa cada letra; para lo segundo habría que *restar* dicha cantidad.
- No existe representación para símbolos de uso frecuente, aunque dependientes del idioma, como en nuestro caso las letras acentuadas, la letra "ñ" o diversos signos de puntuación del español, catalán, francés...

Observa que cada carácter emplea 7 bits para su codificación. Por tanto, es posible representarlo como un valor entre 0 y 127, es decir, ocupa un byte (aunque sobra un bit). Recuerda que, para el ordenador, "lo natural" es trabajar con grupos de bits en *múltiplos de 8* (8, 16, 32, 64). Siete bits no es una agrupación "natural", por lo que se redondea a un byte. Así, una página de texto mecanografiado ocupa unos 2 KB (viene a contener unos 2.000 caracteres).

Ejercicio 11. Contesta a la siguiente pregunta:

> ¿Cuántas páginas de texto mecanografiado caben en un CD-ROM? Para responder, ten en cuenta que un CD-ROM permite albergar unos 700 MB.

Tal como hemos visto, el problema fundamental de la tabla ASCII y la causa principal de que realmente no se utilice tal cual, es que muchos lenguajes usan símbolos que no pertenecen a la tabla ASCII. Es por ello que, tomándola como base, se desarrollaron muchas *extensiones estándar* que agrupan símbolos de idiomas que pertenecen a la misma familia o que son afines entre sí.

Las que más nos interesan son las extensiones *ISO-8859-1*, también conocida como *Latin1* por el hecho de codificar símbolos pertenecientes a lenguas romances occidentales (español, catalán, italiano, francés...), e *ISO-8859-15*, que incluye, además, el símbolo del *euro*. Hay otras ISO-8859 que codifican símbolos pertenecientes a otras "familias" de idiomas. Todas tienen en común que usan *el bit adicional* que se desperdiciaba con la tabla ASCII (con lo que cada carácter se codifica ahora con una secuencia de 8 bits y, por tanto, sigue "cabiendo" en un byte).

3.4.2. Texto

Realmente, codificar un texto no presenta demasiados problemas. Básicamente, un texto es una *sucesión de caracteres*. El único problema que se presenta es determinar cuándo termina la ristra de bits que representa el texto, es decir especificar su longitud. Para ello, se han adoptado, tradicionalmente, dos soluciones:

- indicar al *principio* la *longitud del texto*,
- utilizar un *símbolo terminador* (por ejemplo, el valor 0, es decir, el carácter NUL).

Ejemplo 15. La palabra "Pepe" se puede representar:

- indicando la longitud en primer lugar (el primer byte es el número 4):

00000100	01010000	01100101	01110000	01100101
Número 4	Letra `P`	Letra `e`	Letra `p`	Letra `e`

- o usando un carácter nulo (NUL), por ejemplo, como terminador:

01010000	01100101	01110000	01100101	00000000
Letra `P`	Letra `e`	Letra `p`	Letra `e`	Terminador (0)

□

Es la segunda solución (usar NUL como marca de fin de texto) la que realmente se utiliza. ¿Por qué? Piensa en lo que ocurriría si tuviésemos que representar un texto de más de 256 caracteres. La primera solución requeriría utilizar 2 bytes en lugar de uno, es decir, requiere un cambio. Sin embargo, la segunda solución es igualmente válida tengamos un texto formado por 2 caracteres o por 20.000.

Una última cuestión interesante. Hay que tener en cuenta qué tratamientos se van a aplicar a las informaciones para elegir su representación interna. Por ejemplo, fíjate que los números en sistema decimal tienen dos posibles representaciones en un ordenador. *Como dato numérico*, el número decimal se codificará en su equivalente binario, utilizando una cantidad prefijada de bits para todo el número. Y *como secuencia de caracteres*, cada cifra del número decimal se codificará según la codificación ASCII (o alguna extensión ISO-8859) utilizando un byte para codificar cada dígito.

Si un número en decimal lo almacenamos en memoria como su número binario equivalente, en general ocuparemos menos espacio que si lo hacemos como texto, empleando la codificación ASCII correspondiente. Además, así se pueden realizar directamente operaciones aritméticas con él. Si no se va a realizar ningún tratamiento numérico con dicho número, almacenándolo como secuencia de caracteres, se pueden realizar directamente operaciones de entrada/salida, sin necesidad de hacer conversiones.

Ejemplo 16. Como dato numérico, el número decimal 1.081.472 se codifica, utilizando 32 bits, en el número binario

`00000000000100001000000010000000,`

pero como secuencia de caracteres, el número 1.081.472 se codifica, en ASCII, empleando la siguiente secuencia de siete códigos binarios (uno para cada dígito)

`00110001 00110000 00111000 00110001 00110100`
`00110111 00110010.`

□

3.4.3. Unicode

Figura 3.5. Un pequeño extracto de la tabla Unicode de codificación de caracteres

A pesar de la aparente solución a los problemas de codificación universal que supuso la utilización de las extensiones estándar (como las ISO-8859), éstas no resultaron ser, precisamente, una panacea.

Si bien los diversos idiomas del mundo consiguen ver satisfecho su deseo de representar en el ordenador sus símbolos propios con dichas extensiones, el problema que se plantea es el del *intercambio de información*. ¿Cómo verá en su ordenador un húngaro nuestro mensaje de correo electrónico codificado mediante Latin1, si él emplea otra extensión ISO-8859 no coincidente con la nuestra?

Lo que ocurre realmente, el problema que hemos de afrontar, es que *emplear 8 bits por carácter resulta insuficiente* para representar los símbolos de uso común en las lenguas naturales. Además, la gran variedad de formatos alternativos al ASCII dificulta el intercambio de información, como acabamos de ver en el ejemplo anterior.

Es por ello que se ha diseñado una codificación de *16 bits* por carácter, la cual permite representar 65.536 (2^{16}) símbolos diferentes: a esta tabla de representación de caracteres, estándar y universal, se le conoce con el nombre de *Unicode*. En la actualidad, nos encontramos en un momento de transición desde las codificaciones de 8 bits a Unicode.

3.4.4. Representación gráfica

Ya hemos visto que podemos representar con 8 bits cada carácter y que el texto se representa como una secuencia de caracteres (y determinando su longitud de alguna manera). Sin embargo, en pantalla los caracteres no aparecen como esas secuencias de bits, sino como *imágenes*.

¿Cómo es que vemos estas secuencias de bits como imágenes? Realmente, la pregunta que nos debemos formular primero es ¿cómo se representan esas imágenes? En la siguiente sección responderemos a estas preguntas.

3.5. IMÁGENES

En un ordenador *todo* se representa con bits. Las imágenes no son una excepción y también se representan mediante bits. Podemos ver una imagen como una *matriz de*

píxeles. ¿Y qué es un *píxel*? Pues un píxel es un *punto de la imagen*. El número de píxeles en ancho y en alto determinan la resolución de la imagen. Cada píxel es capaz de almacenar un *valor*: el color que presenta la imagen en dicho punto. En el caso más sencillo, dicho valor es un 0 o un 1, es decir, coincide con el valor de un único bit: esto posibilitaría una imagen en blanco y negro (puro). La Figura 3.6 muestra un ejemplo al respecto.

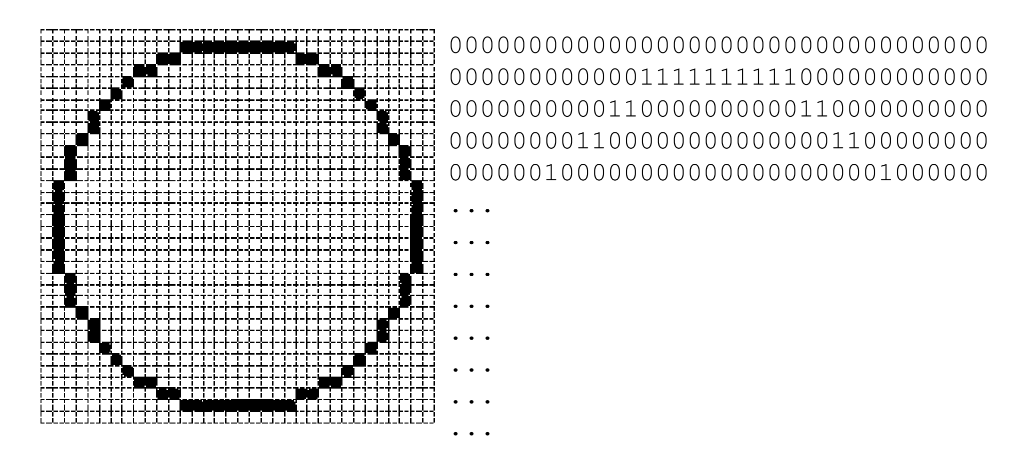


Figura 3.6. Un mapa de bits (bitmap) que representa una imagen en blanco y negro. A la izquierda, la imagen; a la derecha, el mapa de bits

---

Ejercicio 12. Contesta a la siguiente pregunta:

¿Cuántos bytes ocupa una imagen en blanco y negro de 800×600 píxeles?

Volvamos ahora a la cuestión acerca de cómo se muestran los caracteres en pantalla: ¿cómo es que vemos las secuencias de bits que representan internamente a los caracteres como imágenes? Para simplificar la cuestión, asumiremos que cada carácter es una imagen en blanco y negro (el razonamiento sería el mismo si fuese en color).

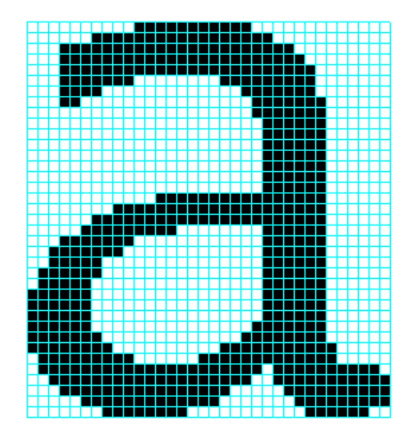


Figura 3.7. La imagen en blanco y negro que representa la letra “a”

Cada carácter a representar (“dibujar”) en pantalla lleva asociada su propia imagen. Todas las imágenes que representan a los caracteres se almacenan en la memoria del ordenador. Una *tabla* se encarga de establecer la correspondencia entre códigos binarios de representación interna e imágenes: indica en qué zona de memoria está guardada la imagen que corresponde a cada letra o carácter identificado mediante su código binario.

Así, por ejemplo, cuando queremos mostrar el carácter correspondiente al código ASCII `01100001`, se muestra en pantalla la imagen que está almacenada en la zona de memoria correspondiente. Los sistemas con tipos de letra más avanzados (los que

se usan en la actualidad) no guardan en memoria una imagen, sino *un programa* capaz de generar esa imagen a diferentes escalas partiendo de una serie de tipos de letra que almacenan imágenes a unos tamaños preestablecidos: es lo que se conoce como tipografías o *fuentes* de letra. Recuerda que el escritorio (tanto en Windows como en Linux) incluye herramientas de configuración del aspecto que permiten cambiar el tipo de letra que se emplea por defecto en el entorno.

Bien, ahora que ya hemos respondido a la cuestión acerca de la representación gráfica de los caracteres en el monitor del ordenador, llega el momento de avanzar más en los conceptos de representación de imágenes propiamente dichos. De momento sólo sabemos representar imágenes en blanco y negro asociando el valor de un único bit por píxel. Por tanto, si queremos representar más colores, tendremos que asociar más información a cada píxel. ¿Cómo? Como casi siempre, aumentando el número de bits en cada píxel. Así, una imagen en blanco y negro con tonalidades de gris necesita varios bits por píxel:

- 1 bits: 2 tonos (blanco y negro).
- 2 bits: 4 tonos.
- 3 bits: 8 tonos.
- ...
- 8 bits (1 byte): 256 tonos.

```
255 254 253 252 ... 3 2 1 0
255 254 253 252 ... 3 2 1 0
255 254 253 252 ... 3 2 1 0
...
```

Figura 3.8. Diversos tonos de gris y su representación correspondiente en forma de mapa de bits

---

Ejercicio 13. Contesta a la siguiente pregunta:

¿Cuántos bytes ocupa una imagen con 256 tonos (niveles) de gris de 800×600 píxeles?

Bien, llegados a este punto, hemos conseguido una mayor riqueza cromática en la representación en blanco y negro. Pero, ¿y si queremos representar colores? Debes tener en cuenta que las imágenes en color se forman combinando tonos de tres colores básicos: rojo, verde y azul (RGB, del inglés “red, green, blue”).

Cada color se describe con tres números que indican el tono de cada color básico. Cada número está entre 0 y 255, es decir, se emplea un byte para describir el tono de cada color básico. Por tanto, describir un color con la codificación RGB requiere 3 bytes por píxel (véase la Figura 3.10). Existen otros “modelos” de color, pero éste es el más extendido.

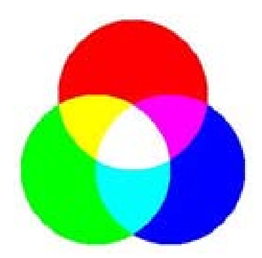


Figura 3.9. Los tres colores básicos en la codificación RGB

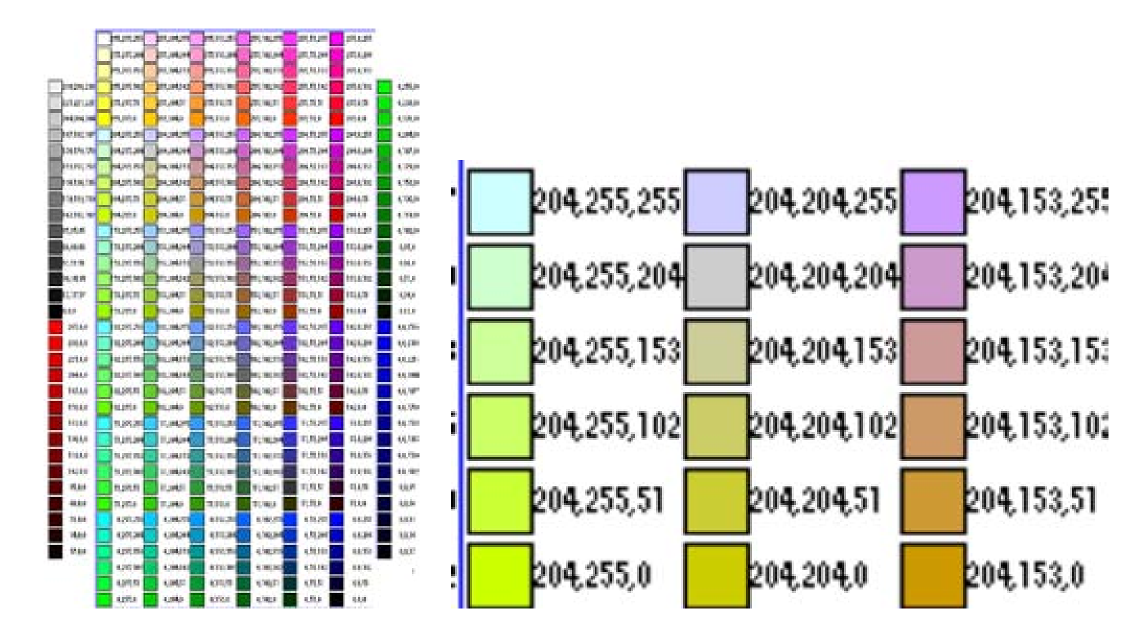


Figura 3.10. Representación de algunos colores mediante la codificación RGB

Ejercicio 14. Contesta a la siguiente pregunta:

¿Cuántos colores pueden representarse con la codificación RGB de 3 bytes?

Ejercicio 15. Contesta a la siguiente pregunta:

¿Cuántos bytes ocupa una imagen de 800×600 píxeles empleando la codificación RGB?

Sin embargo, y aunque la memoria resulta barata, emplear 3 bytes por cada píxel para codificar una imagen en color puede ser un desperdicio de memoria si una imagen usa pocos colores en realidad. Existe un "truco" para solucionar este problema: usar un modelo de color que se conoce con el nombre de *paleta de colores*. La paleta de colores sirve para registrar el uso de un número determinado de colores, establecido a priori, en función de la memoria que deseemos (o podamos) gastar para estos menesteres.

Por ejemplo, si decidimos emplear un byte por píxel (reduciendo a la tercera parte el tamaño de las imágenes en color) entonces serán 256 los colores que podremos usar en la imagen:

índice	**R**	**G**	**B**	*color*
0	255	242	12	
1	242	65	22	
2	200	4	120	
3	1	0	42	
⋮	⋮	⋮	⋮	⋮
255	42	0	255	

En nuestro ejemplo, la imagen necesita ahora un byte por píxel en lugar de los tres anteriores: el valor de dicho byte indica el índice en la tabla que referencia a un color en concreto.

3.6. SONIDO

El sonido es una onda de presión a lo largo del tiempo. Podemos "capturar" sonido con un micrófono que registra la variación de una señal eléctrica a lo largo del tiempo, o a través de una conexión analógica de la tarjeta de sonido (conectándola a la salida AUX en, por ejemplo, tocadiscos, cassettes, radios...) o bien grabando audio digital directamente (mediante conversión de formatos, por ejemplo de CD-Audio a MP3).

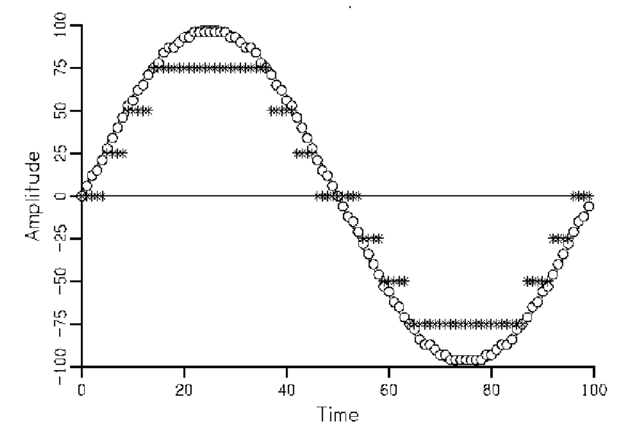


Figura 3.11. Forma de onda del sonido: magnitud continua

La amplitud de la señal de audio es una magnitud continua. Para almacenarla en el ordenador es necesario *digitalizarla* o *discretizarla*, es decir, convertirla en una secuencia de valores enteros. Para ello, se necesitan dos acciones: *muestreo* y *cuantización*. La primera consiste en definir el *número de veces* que la señal va a ser analizada (muestreada) por unidad de tiempo. La segunda implica definir la *escala de valores*, esto es, el rango, que vamos a ser capaces de representar.

Como el lector avispado sin duda habrá captado, esto último tiene que ver con el número de bits que definamos para nuestra escala, a mayor número de bits, mayor rango de valores seremos capaces de codificar y, por tanto, más precisa resultará nuestra medición. Por otro lado, el número de muestras cuantizadas por unidad de tiempo es un parámetro que influye en lo fiel que será nuestra representación del sonido. Cuantos más puntos tenga para interpolar la curva real del sonido, mejor será dicha interpolación, es decir, más realista será nuestro "dibujo" aproximado de la onda acústica. En resumen, hay dos factores que determinan la calidad del sonido digitalizado:

- la cantidad de bits (la resolución) que use para representar el sonido, es decir el número de niveles que tengo para interpolar valores en la cuantización y
- la cantidad de valores (muestras) medidos por segundo (unidad de tiempo), es decir, la *frecuencia de muestreo*, que se mide en herzios (Hz)[7].

Para que te hagas una idea, la música que se "consume" actualmente (en formato CD-Audio) es sonido digital y para nosotros tiene mucha calidad. Pues bien, una grabación con calidad de CD requiere:

- dos formas de onda (sonido estéreo) con 16 bits de resolución para la cuantización ($2^{16} = 65.536$ niveles) y
- tomar 44.100 muestras por segundo (aproximadamente 44 KHz).

Esto implica que una canción de entre 3 y 4 minutos vendría a ocupar entre 30 y 40 MB (44.100 muestras $\times$ 2 (estéreo) $\times$ 180 segundos $\times$ 2 bytes por muestra $=$ $31.752.000$ bytes, que son algo más de 30 MB).

3.7. VÍDEO

Lo primero que tenemos que definir es el concepto de *vídeo*. Por vídeo entendemos una *secuencia de frames* (imágenes estáticas) en *movimiento*, tal como ocurre en el mundo real. La diferencia con una animación (también una secuencia de imágenes en movimiento) es el concepto de *movimiento real*: la secuencia de imágenes tiene que mostrarse lo suficientemente rápido como para constituir una representación convincente del movimiento que se da en la naturaleza.

Esto se consigue a partir de la denominada *frecuencia de fusión*. La retina del ojo del ser humano tiene la propiedad de retener durante unos instantes la imagen. La sensación de movimiento continuo se produce, en el ser humano, por el hecho de que se presenta la siguiente imagen *antes* de que haya desaparecido por completo de la retina la imagen anterior. Se denomina frecuencia de fusión a la cantidad de imágenes que se ha de mostrar por segundo al ojo humano para que se produzca este fenómeno. La frecuencia de fusión es de 40 imágenes por segundo. Cualquier velocidad igual o superior producirá el efecto deseado. Un comentario: en las películas se usan 24 frames/s., pero ten en cuenta que los proyectores muestran dos veces cada frame produciendo efectivamente una tasa de 48 frames/s. Hay dos estándares para el vídeo digital:

(A) NTSC (EE.UU. y Japón); tamaño frames: 640×480, 24 bits de color por píxel y $29{,}97$ frames/s. ($59{,}94$ en la práctica).

(B) PAL (Europa occidental y Australia); tamaño frames: 768×576, 24 bits de color por píxel y 25 frames/s. (50 en la práctica).

Estos son los estándares que se aplican en televisión, transmisión por cable y vídeo digital (DVD, por ejemplo). Podemos "capturar" vídeo de varias fuentes. Por ejemplo,

[7] 1 Hz significa una vez por segundo.

usando una entrada analógica para la conexión de un reproductor VHS, mediante una tarjeta de televisión en el ordenador o, mejor, utilizar una cámara de vídeo digital y conectarla al ordenador para almacenar el vídeo filmado.

Para almacenar un vídeo o una película en un fichero se ha de utilizar un *contenedor*. Un contenedor es como un "sobre" o envoltorio que comprende todas las secuencias de audio (puede haber varias pistas de audio), vídeo y, posiblemente, subtítulos, que conforman un vídeo en un *único fichero*. ¿Por qué es necesario utilizarlo?

- para sincronizar perfectamente audio, vídeo y, tal vez, los subtítulos,
- para poder realizar rebobinado y marcha hacia adelante y hacia atrás,
- para hacer búsquedas rápidas en el fichero,
- para reproducir a partir de un minuto dado,
- para seleccionar (posibles) pistas alternativas de audio,
- para seleccionar subtítulos alternativos,
- para recuperar errores (frames "en mal estado") que puedan impedir la normal reproducción del vídeo,
- ...

A continuación, describimos brevemente cuáles son los formatos o tipos de contenedores de vídeo en ordenador más usados hoy en día:

- AVI (Audio Video Interleaved). Fue el primer formato de vídeo de Microsoft. Está muy extendido en la actualidad, tanto para almacenar vídeos en el ordenador (Windows y Linux), como para *streaming*[8] y descarga en Internet. La información sobre el índice de contenidos (cabecera) se almacena al final del fichero. Tiene un soporte limitado para códecs (compresor/descompresor, como veremos posteriormente). Al ser un formato "cerrado" (consulta el final de este capítulo), ha sido tan "hackeado", es decir, ha sufrido tantas implementaciones por parte de personas ajenas a Microsoft, que hoy en día hay bastantes AVI no conformes al formato originalmente definido.
- MOV. El utilizado por Apple para su famoso QuickTime. Soporta múltiples secuencias de audio y vídeo, el índice de contenidos (cabecera) está al principio del fichero, soporte para pistas de texto y anotaciones, soporte para cientos de códecs, etc.
- MPEG (Motion Picture Experts Group). Es el contenedor empleado para almacenar vídeo codificado con los códecs MPEG-1 (VCD) y MPEG-2 (DVD). Normalmente se mezcla con una pista de audio.
- OGG. Similar al AVI, soporta los mismos códecs y, además, OGG Vorbis (formato de compresión de audio con características similares a MP3 y libre de patentes).

[8]Reproducción on-line mediante el plugin de un navegador o programa específico, tipo Real Player.

3.8. COMPRESIÓN

A estas alturas, ya hemos visto que es posible representar cualquier tipo de información empleando bits o, mejor dicho, grupos de bits. Sin embargo, también hemos apreciado que la representación directa de la información puede ocupar mucha memoria. Sobre todo cuando hablamos de contenidos multimedia: imágenes, vídeo y sonido.

En la actualidad, se han diseñado muchas técnicas de *compresión de la información*, cuyo objetivo consiste en reducir en gran medida la ocupación de memoria. Todas estas técnicas se basan en un principio fundamental de diseño para lograr este objetivo: *aprovechar la redundancia* de la información. Precisamente para ilustrar este importante concepto de redundancia en la información, vamos a abordar un ejemplo de técnica de compresión primitiva, desarrollada hace ya mucho tiempo: *run-length encoding*.

3.8.1. Run-length encoding

Run-length encoding es una técnica que se desarrolló para comprimir imágenes en blanco y negro (mapas de bits o *bitmaps*). En lugar de almacenar los bits de la imagen, almacenamos, para cada fila, el número de bits de cada color. La Figura 3.12 ilustra su funcionamiento. Analiza dicha figura y observa cómo se aprovecha la redundancia en la información (hay series consecutivas con el mismo valor del bit: 0 ó 1) para reducir el tamaño de la imagen original en la compresión. Por ejemplo, en la primera fila tenemos 34 bits en blanco; en la segunda 12 bits en blanco, seguidos de 10 en negro y otros 12 en blanco; y así sucesivamente.

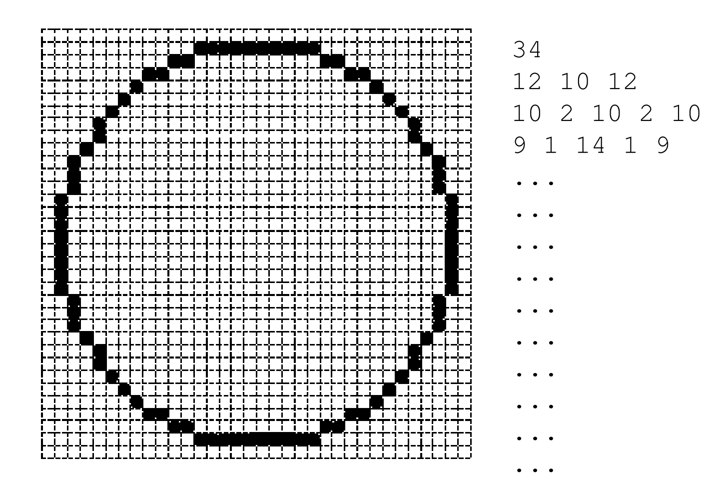


Figura 3.12. Así actúa *run-length encoding*

3.8.2. Compresión con y sin pérdidas

La técnica que acabamos de ilustrar, *run-length encoding*, permite "recuperar" la información original a partir de la forma comprimida tal cual: es una técnica de *com-*

presión sin pérdidas, puesto que no se descarta ningún tipo de información a la hora de comprimir.

Sin embargo, muchas de las técnicas de compresión empleadas en la actualidad *descartan información* al comprimir. No es posible recuperar esta información al descomprimir. Se dice que son técnicas de *compresión con pérdidas*. Cabe preguntarse si esto es asumible. Normalmente, comprimimos la información para ahorrar espacio en disco, por lo que, una vez comprimida, borramos la información original. ¿Qué sentido tiene, pues, utilizar estas técnicas?

La respuesta es que, de hecho, la información que se descarta carece de sentido para el ser humano en la mayoría de las ocasiones. Nuestras imperfecciones hacen que, o bien no podamos captarla o bien no nos resulte necesaria para "entender" la información como un todo. Las técnicas de compresión con pérdidas son, sobre todo, muy útiles en la compresión de imágenes, sonido y vídeo; no permiten recuperar la información original, pero, por ejemplo:

- La compresión JPEG permite almacenar imágenes ocupando mucho menos espacio gracias a que el ojo humano no afina mucho en la distinción de variaciones de color.
- La compresión MP3 comprime el sonido haciendo uso de información de carácter psicoacústico: no oímos sonidos de frecuencias próximas, ante un sonido fuerte los más débiles quedan ocultos, etc.

Para que te hagas una idea de lo que estamos hablando. La compresión en vídeo digital resulta fundamental. Echa cuentas: un segundo de vídeo en formato NTSC viene a ocupar unos 26 MB, por tanto, un minuto son aproximadamente $1{,}6$ GB. Otro tanto ocurre con el sistema PAL: 1 segundo de vídeo en este formato viene a ocupar 31 MB, por tanto, un minuto son $1{,}85$ GB.

Para comprimir vídeo se utilizan los llamados *códecs* (acrónimo inglés del equivalente castellano *compresor/descompresor*). La mayoría se basan en el almacenamiento único de las partes invariables comunes a todos los frames que constituyen una secuencia, junto con índices que permiten reconstruir los (pequeñísimos) cambios de un frame a otro.

Todos tienen en común que descartan algún tipo de información al comprimir, por lo que podría llegar a apreciarse una disminución en la calidad de la imagen. Pero si se usan los adecuados *niveles de muestreo y cuantización* (frecuencia y número de bits) la pérdida es *inapreciable* para el ser humano. Recuerda: niveles más altos implican más calidad, pero también menos compresión. Veamos algunos de los códecs más utilizados en la actualidad y, además, compatibles con el contenedor AVI:

- DivX, basado en el estándar MPEG-4, es capaz de comprimir una película en DVD hasta 1/10 parte de su tamaño sin apenas merma de calidad. Para obtener una calidad de imagen similar a la del DVD, se pueden comprimir aproximadamente 90 minutos de película en un CD (700 MB), si, además, quieres conservar sonido tipo Dolby, entonces sólo serían 75 minutos aproximadamente los que cabrían en un CD (con calidad de imagen similar a DVD). Es el formato "de facto" para el intercambio de películas en Internet. Es al vídeo digital (DVD) lo que el MP3 a la música.

- Xvid (DivX al revés) es la implementación *libre y gratuita* de MPEG-4. Tiene prestaciones iguales o superiores a DivX. Cada vez se usa más en Internet (para películas).
- ffmpeg4…

Recuerda que MPEG, como contenedor, también usa códecs, pero éstos deben ser los propios de MPEG (MPEG1, MPEG2). Bien, terminamos el asunto de la compresión abriendo un par de reflexiones:

1.- ¿Qué crees que pasaría si usásemos una técnica de compresión con pérdidas al texto de El Quijote?

    Que más vale que no tirásemos el original de la novela. Ciertamente, la versión comprimida ocuparía menos espacio, pero ya no tendríamos la novela ni forma de recuperar el original a partir de la versión comprimida.

    ¿Obtendríamos una versión resumida del libro?

    No. Las técnicas de compresión *no analizan la semántica* para comprimir. Eso sería propio de la Inteligencia Artificial (y no es un problema resuelto). Simplemente buscan *patrones repetidos* y *eliminan información redundante* de carácter "estructural", o bien descartan información considerada superflua.

2.- ¿Qué pasaría si usásemos repetidamente una técnica de compresión sin pérdidas al texto de El Quijote?

    Llegaría un momento que no podríamos comprimir más: *existe un límite para la capacidad de compresión*.

    ¿Sería posible llegar a almacenarlo en un único bit?

    Como hemos dicho, hay un límite. El mínimo número de bits necesario para codificar una información *sin pérdidas* guarda relación con el concepto de *entropía*: cuanto más orden, más capacidad de predicción y menos bits; a más desorden, menos predecible será la secuencia y necesitaremos más bits para codificar la compresión.

3.9. FORMATOS DE FICHEROS

En resumen, todos los ficheros del ordenador codifican la información en binario, es decir, como secuencias de ceros y unos. No obstante, distinguimos *grosso modo* dos grandes tipos de ficheros:

- Ficheros *de texto*: codifican secuencias de caracteres en ASCII (o alguna extensión estándar como Latin1 de las ISO-8859) o Unicode. Pueden leerse con herramientas convencionales y su contenido puede modificarse con editores de texto.

- Ficheros *binarios* propiamente dichos: son ficheros cuyo contenido no puede visualizarse directamente, pues no está descrito con texto. Siguen ciertas convenciones "especiales" definidas por sus creadores (formato). Se necesitan herramientas especiales para modificar su contenido (normalmente desarrolladas por los creadores del formato).

Tanto los ficheros de texto como los ficheros binarios se pueden clasificar, además, por el *tipo de contenido* que almacenan. Es costumbre que los diferentes tipos de fichero se "marquen" con la extensión del nombre que los identifica. Veamos una enumeración de los tipos de fichero más usuales en el trabajo cotidiano junto con sus extensiones más frecuentes (por costumbre):

- Texto propiamente dicho (`txt`): fichero de texto convencional.
- Imágenes (`tiff`, `bmp`, `gif`, `jpg`, `jpeg`, `png`...): las hay sin (`bmp`) y con compresión (como `jpeg` o `png`).

Formatos de imagen

Los factores que afectan al espacio en disco que ocupará el archivo que contiene una imagen son los dos vistos con anterioridad: *número de píxeles* (tamaño de la imagen) y *modelo de color* (blanco y negro, grises, color RGB, paleta de colores). La mayoría de los formatos de imagen aplican compresión (con o sin pérdidas):

BMP (bitmap): almacena la imagen tal cual, sin comprimir. Es el formato descrito con anterioridad en la sección 3.5.

GIF (Graphics Interchange Format): usa un esquema de compresión sin pérdidas LZW (Lempel-Ziv-Welch). Resulta apropiado para imágenes con bordes muy marcados y grandes áreas de colores planos (por ejemplo, imágenes con textos y dibujos basados en líneas de forma predominante). Modelo de color: *paleta de colores*; sólo 8 bits por píxel, es decir, puede indexar 256 colores distintos, por lo que debe tenerse en cuenta la pérdida de calidad si la imagen original representaba más de 256 colores. Permite visualización progresiva (entrelazado), adecuado para imágenes en web. Permite animaciones, ya que se puede almacenar más de una imagen en un único fichero.

JPEG (Joint Photographic Experts Group): resulta apropiado para imágenes con transiciones de color sutiles y suavizadas (fotografías e imágenes en escala de grises o con millones de colores). Realiza una compresión con pérdidas; se debe buscar un equilibrio entre la ocupación y la calidad de la imagen. El método de compresión trocea la imagen en pequeños bloques y promedia el color en cada bloque: se basa en que el ojo humano no afina mucho en la distinción de variaciones de color (esto hace que el texto y los dibujos basados en líneas no sean apropiados para JPEG). Permite hasta 24 bits de color por píxel. El formato JPEG progresivo permite entrelazado (el estándar, no).

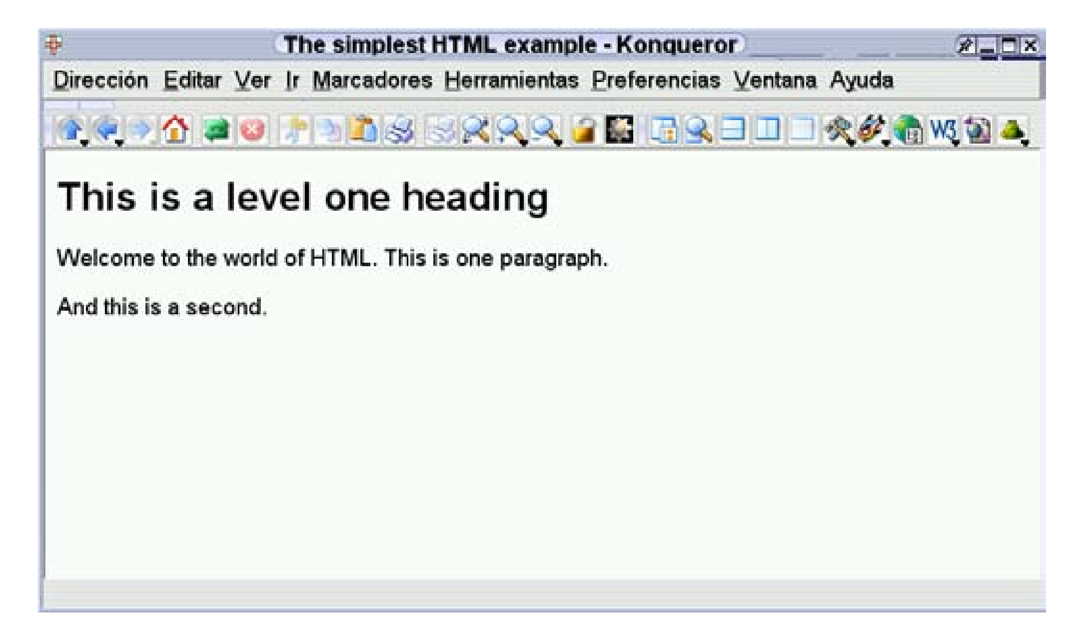
continúa en la página siguiente ...

. . . viene de la página anterior

PNG (Portable Network Graphics): resulta apropiado para cualquier tipo de imagen. Utiliza el método de compresión “Deflate”, similar al LZW; pero aumentando el grado de compresión. Permite hasta 48 bits de color por píxel y realiza una compresión sin pérdidas (al contrario que JPEG), lo que, junto al ilimitado número de colores (al contrario que GIF), proporciona gran calidad con cualquier tipo de imagen. También permite entrelazado.

- Sonido (`wav`, `mp3`, `ogg`. . .): los hay sin compresión (como `wav`) y con compresión (como `mp3` u `ogg`).
- Vídeo (`mpg`, `mpeg`, `avi`, `mov`. . .). Normalmente la extensión hace referencia al contenedor y no al códec empleado para comprimir el vídeo propiamente dicho.
- Fichero imprimible en impresora postscript (`ps`).
- Documento portable (`pdf`). Es propiedad de la empresa Adobe, pero es estándar y públicamente conocido (“abierto”).
- Documentos generados por aplicaciones con formato binario:
  - Microsoft Word (`doc`).
  - Microsoft Excel (`xls`).
  - Photoshop (`psd`).
  - Gimp (`xcf`).
  - . . .
- Ficheros de texto aptos para visualización en web (`htm`, `html`).
- Ficheros de texto aptos para el formateador de documentos TEX(`tex`).

Observa que algunos tipos de fichero siguen siendo ficheros de texto. Aquí tienes un ejemplo: esta página web



muestra, en realidad, el contenido de este fichero de texto:

```
<HTML>
 <HEAD>
  <TITLE>The simplest HTML example</TITLE>
 </HEAD>
 <BODY>
  <H1>This is a level one heading</H1>

  <P>Welcome to the world of HTML. This is one paragraph.</P>

  <P>And this is a second.</P>
 </BODY>
</HTML>
```

Pero aunque el contenido sea texto, sigue unas normas precisas (lenguaje HTML) que permiten al navegador mostrarlo con un formato visual atractivo. Cada tipo de fichero tiene programas que permiten su edición y/o visualización. Algunos tipos de fichero disponen de visualizadores específicos:

- HTML: los navegadores web (Netscape, Mozilla, Konqueror, Internet Explorer...).
- PDF: Acroread y xpdf.
- JPEG, GIF, PNG: programas de dibujo, `xv`, `kuickshow`, `ee`...
- MPG, AVI: programas multimedia como `mplayer`, `xine`...
- Postscript: Ghostview, gv.
- ...

Formatos de sonido

Los factores que afectan al espacio en disco que ocupará el archivo que contiene sonido (y a la calidad del mismo) son los dos vistos con anterioridad: frecuencia de muestreo y niveles de cuantización (número de bits). La mayoría de los formatos de sonido aplican compresión (con pérdidas):

CDA (Compact Disc Audio): es el formato empleado en los CD-Audio (música). Gran calidad: 44 KHz, 16 bits, estéreo. Inconveniente: ocupa mucho espacio.

WAV: desarrollado por Microsoft para la reproducción y edición de audio en PC; misma calidad que CDA; mismo inconveniente (espacio ocupado).

MP3 (Capa III de MPEG-2, Moving Picture Experts Group): compresión con pérdidas. Se basa en que no percibimos sonidos de frecuencias próximas y que ante un sonido fuerte los más débiles quedan ocultos (características psico-acústicas) para descartar información que no es valiosa para el ser humano. Frecuencia estándar: 44 KHz (mejor no tocar) y con la *tasa de bits* o *bitrate* controlamos la relación entre calidad del sonido y espacio que ocupa el fichero comprimido.

continúa en la página siguiente ...

. . . viene de la página anterior

Los expertos dicen que a un bitrate de 160 kbps. el ser humano no distingue entre original y MP3. Se necesita un reproductor con soporte MP3 para poder escuchar estos ficheros. Hoy en día prácticamente cualquier aparato con capacidad de reproducir audio lo soporta. Está patentado (se paga licencia). Se usa como estándar “de facto” para el intercambio de ficheros en Internet.

OGG Vorbis: formato de compresión con pérdidas de audio, más evolucionado que MP3 y libre (público). Sus especificaciones son de *uso gratuito*, incluso si se reproduce con software de pago. Usa principios matemáticos distintos a MP3, pero la idea básica es la misma: usar modelos humanos de audición para eliminar la información que el oído humano es incapaz de captar. Como en MP3 el *bitrate* controla el compromiso calidad del audio/espacio ocupado. Obtiene una calidad de sonido similar a MP3 (en algunos casos mejor) y ocupa, poco más o menos, el mismo espacio. También se usa para intercambio de ficheros en Internet.

Más sobre sonido

Para reproducción: Windows Media Player (Windows) y XMMS (multiplataforma) reproducen todos los mencionados: CD-Audio, WAV, MP3, OGG y otros.

Existen reproductores de bolsillo que soportan MP3 (y muchos también OGG).

La gran mayoría de DVD electrodomésticos soportan MP3 (y algunos empiezan a incluir OGG).

Conversión entre formatos:

CDA-WAV: Programas de “ripeo”, Easy CD Ripper (Windows), Grip (Linux). . .

WAV-MP3: la mayoría de programas de “ripeo” incluyen una herramienta para efectuar esta conversión. También hay herramientas específicas como LAME (Linux).

WAV-OGG: la mayoría de programas de “ripeo” incluyen una herramienta para efectuar esta conversión, p.e. Grip.

WAV-CDA: se necesita un programa de grabación de CD como Nero Burning (Windows) o K3b (Linux). Seleccionar los ficheros y escoger “grabar CD de audio”.

MP3,OGG-CDA: los buenos programas de grabación de CD hacen la conversión de formato automática a CD-Audio, p.e. Nero (Windows) y K3b (Linux).

Para editar audio: Adobe Premiere. SW. libre: Jahshaka (multiplataforma).

A tener en cuenta. . .

Herramientas de conversión de vídeo: para pasar de DVD a DivX, Xvid, VCD, SVCD. . . Hay programas como DVD Rip N’ Burn (Windows, www.dvdripnburn.com) y dvd::rip (Linux, www.exit1.org/dvdrip/).

continúa en la página siguiente . . .

. . . viene de la página anterior

Ya hay bastantes reproductores de DVD (electrodomésticos, ¡y a buen precio!) que soportan DivX/Xvid y son actualizables por Internet.

3.9.1. Formatos abiertos y cerrados

Los ficheros en un formato determinado se atienen a unas reglas precisas que indican qué se puede almacenar en ellos y cómo se estructura la información que contienen.

- Los *formatos abiertos* siguen normativas públicas y que puede seguir cualquiera.

  Usar formatos abiertos hace posible que se comparta información entre diferentes usuarios sin coste alguno.

- Los *formatos cerrados* siguen normativas que fijan empresas y que se mantienen en secreto.

  Mantener un formato cerrado puede suponer una ventaja competitiva, sobre todo cuando el formato se convierte en un estándar de facto en el intercambio de información: *los usuarios se ven obligados a adquirir productos comerciales para poder leer o modificar los ficheros.*

Debes tener muy presente que utilizar un formato cerrado para el intercambio de información puede resultar contraproducente, ya que estás obligando al receptor a:

(a) adquirir una licencia del programa, con el consiguiente dispendio, o a

(b) obtener una copia ilegal del programa, lo cual puede conllevar a su ingreso en prisión, con el nuevo código penal en la mano (en vigor desde el 1 de octubre de 2004).

SOLUCIONES A LOS EJERCICIOS PROPUESTOS

Ejercicio 1. ¿Qué valores decimales representan los siguientes números binarios?

a) 1111_2 representa el 15.

b) 1_2 representa el 1.

c) 0_2 representa el 0.

d) 0111_2 representa el 7.

e) 10_2 representa el 2.

f) 1000_2 representa el 8.

Ejercicio 2. Responde a estas cuestiones:

a) ¿Cuál es el valor más grande que podemos expresar con cuatro bits?

15; los 4 bits a 1: 1111.

b) ¿Y con ocho?

255; los 8 bits a 1: 11111111.

c) ¿Y con n, siendo n un entero positivo?

$2^n - 1$. Observa que con n bits representamos 2^n valores distintos. Si el más pequeño es el 0, entonces, el más grande tiene que ser $2^n - 1$. Otra forma de verlo: una secuencia de n bits todos a 1 es $2^{n-1} + 2^{n-2} + 2^{n-3} + \ldots + 2^2 + 2^1 + 2^0 = \sum_{i=0}^{n-1} 2^i = 2^n - 1$.

Ejercicio 3. Convierte los siguientes números decimales a base 2:

a) 0 se representa en binario como 0.

b) 1 se representa en binario como 1.

c) 18 se representa en binario como 10010.

d) 128 se representa en binario como 10000000.

e) 127 se representa en binario como 1111111.

f) 863 se representa en binario como 1101011111.

g) 2^{10} se representa en binario como 10000000000 (observa que, en este caso, no haría falta hacer ninguna división, ya que el número decimal viene expresado como una potencia de dos: basta con poner un uno seguido de 10 ceros).

Ejercicio 4. Calcula las siguientes sumas de números binarios:

a) $101011_2 + 100101_2$

b) $1_2 + 1001_2$

c) $10000_2 + 10000_2$

d) $101010_2 + 10101_2$

e) $101010_2 + 101010_2$

```
  1   1 1 1 1              1              1
    1 0 1 0 1 1                  1          1 0 0 0 0
+   1 0 0 1 0 1      +     1 0 0 1      +   1 0 0 0 0
  1 0 1 0 0 0 0            1 0 1 0        1 0 0 0 0 0
```

```
                               1     1     1
    1 0 1 0 1 0                  1 0 1 0 1 0
+     1 0 1 0 1             +    1 0 1 0 1 0
--------------             ----------------
    1 1 1 1 1 1               1 0 1 0 1 0 0
```

Ejercicio 5. Efectúa las siguientes operaciones con números binarios:

a) $10011_2 - 101_2$

b) $1001_2 \times 1001_2$

c) $110000_2/10_2$

```
   1 0 0 1 1                      1 0 0 1
    1 1                         × 1 0 0 1
-        1 0 1             ----------------
-------------                     1 0 0 1
   0 1 1 1 0                    1
                                0 0 0 0
                              0 0 0 0
                           +  1 0 0 1
                           ----------------
                              1 0 1 0 0 0 1

       1 1 0 0 0 0 | 1 0
     - 1 0         |------
     -----           1 1 0 0 0
       0 1 0
       - 1 0
       -----
         0 0 0 0 0
```

Ejercicio 6. Convierte a base 10 estos números binarios en complemento a uno de cuatro bits:

a) 1101 es negativo (el bit más significativo es 1), por tanto invertimos los bits: 0010, que es el número 2. Por consiguiente, tenemos -2.

b) 0101 es positivo (el bit más significativo es 0), por tanto: $2^2 + 2^0 = 5$.

c) 0000 es positivo (el bit más significativo es 0), y es claramente el número 0.

d) 1111 es negativo (el bit más significativo es 1), por tanto invertimos los bits: 0000, que es el número 0. Luego, fíjate que en complemento a 1 hay dos formas distintas de representar el 0.

Ejercicio 7. Responde a esta pregunta:

¿Cuáles son los números mayor y menor que podemos expresar en complemento a uno de cuatro bits?

El número mayor será un número positivo, por tanto, el bit más significativo será 0. El valor más grande vendrá determinado al colocar los restantes 3 bits a 1, así: 0111, que es el número 7.

El número menor será negativo, por tanto, el bit más significativo será 1. El valor más pequeño vendrá determinado al colocar los restantes 3 bits a 0, así: 1000, que es el número -7 (fíjate en que invertimos los bits del de arriba).

Ejercicio 8. Efectúa las siguientes operaciones codificando los números en base 10 a sus equivalentes en complemento a uno con cuatro bits:

a) $3 + 2$ es $0011 + 0010 = 0101$.

b) $3 - 2$ es $0011 + 1101 = 0000$ y acarreo 1, por tanto, sumamos 1 y resulta 0001.

c) $1 + 1$ es $0001 + 0001 = 0010$.

d) $1 - 1$ es $0001 + 1110 = 1111$ que equivale a 0000 (0).

e) $7 - 7$ es $0111 + 1000 = 1111$ que equivale a 0000 (0).

f) $-2 - 2$ es $1101 + 1101 = 1010$ y acarreo 1, por tanto, sumamos 1 y resulta 1011 (que como puedes comprobar es -4 en complemento a 1: 0100, si invertimos los bits es 4).

g) $-2 + 3$ es $1101 + 0011 = 0000$ y acarreo 1, por tanto, sumamos 1 y resulta 0001.

Ejercicio 9. Efectúa las siguientes operaciones en complemento a 2 con 4 bits:

a) $3 + 2$ es $0011 + 0010 = 0101$.

b) $3 - 2$ es $0011 + 1110 = 0001$ (el acarreo final 1 se desprecia).

c) $1 + 1$ es $0001 + 0001 = 0010$.

d) $1 - 1$ es $0001 + 1111 = 0000$ (el acarreo final 1 se desprecia).

e) $7 - 7$ es $0111 + 1001 = 0000$ (el acarreo final 1 se desprecia).

f) $-2 - 2$ es $1110 + 1110 = 1100$ (el acarreo final 1 se desprecia). Observa que 1100 es -4, ya que, si le cambiamos el signo, restándole 1 $(1100 - 0001 = 1011)$, e invirtiéndole los bits resultantes, tenemos 0100 que es 4.

g) $-2 + 3$ es $1110 + 0011 = 0001$ (el acarreo final 1 se desprecia).

Ejercicio 10. Codifica en binario los siguientes números en base 10:

- $10{,}25$: parte entera, 10 es 1010; parte decimal es $0{,}25 \times 2 = \mathbf{0}{,}50 \times 2 = \mathbf{1}{,}0$, por tanto, 01. Resultado: $1010{,}01$.
- $128{,}125$: parte entera, 128 es 10000000; parte decimal es $0{,}125 \times 2 = \mathbf{0}{,}25 \times 2 = \mathbf{0}{,}50 \times 2 = \mathbf{1}{,}0$, por tanto, 001. Resultado: $10000000{,}001$.

- $1{,}75$: parte entera, 1 es 1; parte decimal es $0{,}75 \times 2 = \mathbf{1}{,}50$; $0{,}50 \times 2 = \mathbf{1}{,}0$, por tanto, 11. Resultado: $1{,}11$.
- $2{,}875$: parte entera, 2 es 10; parte decimal es $0{,}875 \times 2 = \mathbf{1}{,}75$; $0{,}75 \times 2 = \mathbf{1}{,}50$; $0{,}50 \times 2 = \mathbf{1}{,}0$, por tanto, 111. Resultado: $10{,}111$.
- $3{,}3$: parte entera, 3 es 11; parte decimal es $0{,}3 \times 2 = \mathbf{0}{,}6 \times 2 = \mathbf{1}{,}2$; $0{,}2 \times 2 = \mathbf{0}{,}4 \times 2 = \mathbf{0}{,}8 \times 2 = \mathbf{1}{,}6$; y queda, nuevamente, como parte decimal $0{,}6$, luego a partir de aquí obtendríamos la misma secuencia de decimales una y otra vez; es decir, hemos encontrado un periodo. Resultado: $11{,}0\overline{1001}$. Observa que, en este caso, una secuencia *finita* de decimales da lugar a una secuencia *infinita* (periódica) en binario.

Ejercicio 11. Contesta a la siguiente pregunta:

¿Cuántas páginas de texto mecanografiado caben en un CD-ROM? Para responder, ten en cuenta que un CD-ROM permite albergar unos 700 MB.

700 MB son, aproximadamente, unos 700.000 KB. Por tanto, si una página ocupa 2KB, entonces en un CD-ROM caben en torno a unas 350.000 páginas de texto mecanografiado. El cálculo exacto sería: $(700 \times 1.024)/2 = 358.400$ páginas.

Ejercicio 12. Contesta a la siguiente pregunta:

¿Cuántos bytes ocupa una imagen en blanco y negro de 800×600 píxeles?

$800 \times 600 = 480.000$ píxeles. Una imagen en blanco y negro requiere un bit por píxel, por tanto, la imagen ocupa 480.000 bits que son 60.000 bytes y que son, aproximadamente, unos 60 KB (exactamente serían $60.000/1.024 = 58{,}59375$ KB).

Ejercicio 13. Contesta a la siguiente pregunta:

¿Cuántos bytes ocupa una imagen con 256 tonos (niveles) de gris de 800×600 píxeles?

$800 \times 600 = 480.000$ píxeles. Una imagen con 256 tonos de gris requiere un byte por píxel, por tanto, la imagen ocupa 480.000 bytes que son, aproximadamente, unos 480 KB (exactamente serían $480.000/1.024 = 468{,}75$ KB).

Ejercicio 14. Contesta a la siguiente pregunta:

¿Cuántos colores pueden representarse con la codificación RGB de 3 bytes?

3 bytes son 24 bits, por tanto, pueden representarse $2^{24} = 16.777.216$ colores distintos.

Ejercicio 15. Contesta a la siguiente pregunta:

¿Cuántos bytes ocupa una imagen de 800×600 píxeles empleando la codificación RGB?

$800 \times 600 = 480.000$ píxeles. Una imagen RGB requiere 3 bytes por píxel, por tanto, la imagen ocupa $480.000 \times 3 = 1.440.000$ bytes que son, aproximadamente, unos $1{,}44$ MB (exactamente serían $1.440.000/(1.024 \times 1.024) = 1{,}373$ MB).

CAPÍTULO 4

HARDWARE DEL PC

BIBLIOGRAFÍA

Con el fin de actualizar los datos recogidos en este capítulo, se ha utilizado información presente en las páginas web de las empresas Intel (`http://www.intel.com`) y AMD (`http://www.amd.com`) en el momento de su redacción. Además, en el texto aparecen citas y referencias a material consultado en la *Wikipedia*.

WIKIPEDIA, LA ENCICLOPEDIA LIBRE: `http://es.wikipedia.org`. Fechas de consulta: septiembre y octubre de 2007, enero y febrero de 2008.

WIKIPEDIA, THE FREE ENCYCLOPEDIA: `http://en.wikipedia.org`. Fechas de consulta: septiembre y octubre de 2007, enero y febrero de 2008.

4.1. ¿QUÉ ES UN ORDENADOR?

Un ordenador es un dispositivo electrónico de propósito general capaz de realizar, básicamente, cálculos aritméticos y operaciones lógicas, y dotado de una memoria que le permite almacenar información. En realidad, puede realizar operaciones más complejas y procesar datos, imágenes, sonido, etc. Gracias a la enorme velocidad a la que es capaz de realizar las operaciones básicas, un ordenador puede recibir todo tipo de información, procesarla y, una vez procesada, emitirla al usuario ya "digerida" para su interpretación. Es, por tanto, una máquina susceptible de procesar datos de entrada y generar una información de salida utilizando una serie de instrucciones (programa) previamente almacenadas.



Figura 4.1. Un ordenador personal

Técnicamente hablando, el ordenador es un conjunto de circuitos electrónicos comprimidos en pastillas de silicio, llamadas *chips*, siendo su función fundamental la de enviar señales electromagnéticas de un dispositivo a otro. La parte fundamental de

un ordenador es el *microprocesador*, el verdadero cerebro electrónico que gobierna el funcionamiento de la máquina. También se le denomina CPU, del inglés *C*entral *P*rocessing *U*nit, Unidad Central de Proceso, en castellano. Existen otros dispositivos electrónicos, tales como las calculadoras electrónicas, que comparten muchas de estas características con los ordenadores. Sin embargo, un ordenador se diferencia de una calculadora:

- por ser capaz de almacenar un programa de computador,
- por el número y la complejidad de las operaciones que puede realizar,
- y por su habilidad para procesar, almacenar y recuperar datos sin necesidad de intervención humana.

En general, existen muchos tipos de ordenadores:

- Tenemos los denominados *súper-ordenadores* (Figura 4.2). Se trata de ordenadores que, generalmente, se encuentran en centros de cálculo y que trabajan conectando varios procesadores en paralelo, ya sea en la propia máquina o bien en un *cluster*[1] de ordenadores conectados en una red local (sistemas distribuidos). Encuentran su aplicación en tareas avanzadas de ingeniería e investigación, tales como meteorología (predicción del tiempo, análisis del clima...), ingeniería (simulación, cálculo de estructuras...), física (análisis de datos enviados por sondas espaciales, simulación de experimentos...), etc.

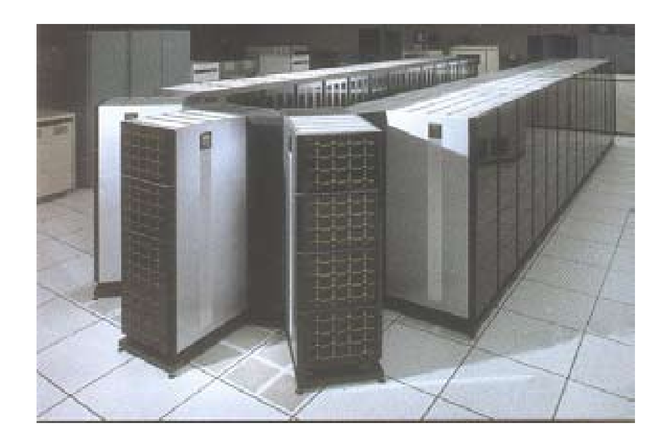


Figura 4.2. Un sistema de computación distribuida en un centro de cálculo

- Igualmente, existen potentes ordenadores denominados *macro-ordenadores* (*en inglés mainframes*, Figura 4.3). Son ordenadores potentes, generalmente multiprocesador y equipados con grandes cantidades de memoria y discos SCSI de altas prestaciones. Son equipos con gran capacidad de procesamiento y capaces de soportar la conexión y el trabajo de numerosos usuarios. Se usan en grandes empresas.

[1] Un *cluster* es una agrupación de ordenadores, físicamente independientes, pero lógicamente conectados entre sí (procesador y memoria compartidas)



Figura 4.3. Un *mainframe* para dar servicio a los usuarios de una gran empresa

- Por otra parte, los *mini-ordenadores*, también denominados servidores (Figura 4.4), son equipos que ofrecen menos prestaciones que los anteriores, aunque suelen estar dotados con uno o dos potentes procesadores, una buena cantidad de memoria RAM (del inglés *Random Access Memory*) y disco(s) de gran capacidad (no SCSI, normalmente SATA-2[2]).



Figura 4.4. Servidor para una pequeña y mediana empresa

Estas máquinas suelen proporcionar servicios a una comunidad de usuarios: por ejemplo, espacio en disco para almacenar información, espacios web personales, buzones de correo, o bien proporcionan servicios de Internet, tales como web, correo electrónico, transferencia de ficheros, etc. También suelen emplearse para gestionar una red local interna (*intranet*) y actuar como punto de acceso a la red exterior (Internet), proporcionando servicios de NAT ("enmascaramien-

[2]Más adelante, en este capítulo, descubriremos el significado de todas estas siglas. De momento, basta con saber que ambos son tipos de discos duros y que se diferencian en precio y prestaciones (siendo SCSI más rápidos y también más caros).

to” de IP), *firewall* (protección contra intrusiones y “ataques” externos) y filtro antivirus. Se utilizan en pequeñas y medianas empresas.

- Para la informática “doméstica” y el uso cotidiano (ofimática, ocio, juegos, navegación, etc.) tenemos los *ordenadores personales* (Figura 4.5). Hoy en día, suelen ser equipos con un procesador potente (tecnología *core duo* o similar), uno o dos GB de memoria RAM, disco duro SATA, tarjeta gráfica 3D (marca ATI o nVIDIA) con 256 MB de RAM (propia), puertos USB 2.0 y/o FireWire (para conexiones de alta velocidad con dispositivos “multimedia” externos, tales como cámaras de vídeo y fotos, o discos externos de gran capacidad) y capacidad de conexión a Internet (módem y/o tarjeta Ethernet integrada en placa).



Figura 4.5. Un ordenador personal con capacidad multimedia

- Los *ordenadores portátiles* constituyen una “clase” especial de ordenadores personales (Figura 4.6). De hecho, ofrecen prestaciones y poseen características similares a éstos, pero están equipados con circuitería específica (procesadores, discos y memoria de bajo consumo y menores dimensiones). Es por ello que su precio es mayor que el del ordenador personal que pueda equipararsele en prestaciones y características. El auge de las redes Wi-fi (la mayoría de los portátiles vienen equipados con tarjetas Ethernet y Wi-fi y este tipo de redes funcionan en bibliotecas, universidades, aeropuertos, etc.), sumado a su mayor autonomía (equipados con baterías de calidad y con la nueva generación de procesadores y circuitería específica de bajo consumo, pueden alcanzar las 4-6 horas de funcionamiento sin conexión a la red eléctrica) lo convierten en una potente herramienta para el profesional que necesita poder trabajar con un ordenador en entornos de trabajo variopintos. Hoy en día, incluso constituyen una alternativa a considerar frente a la adquisición de un ordenador de sobremesa: se pueden obtener prestaciones similares por algo más de precio, pero con la ventaja de la movilidad.

- También tenemos otro tipo especial; los *ordenadores de mano* o *agendas electrónicas*, también conocidas como PDA (del inglés *Personal Digital Assistant*, Figura 4.7). Los asistentes personales digitales son computadores de mano, originalmente diseñados como agendas electrónicas (con calendario, lista de contactos, bloc de notas y recordatorios, citas, alarmas), equipados con pantalla táctil y, generalmente, con un sistema de reconocimiento de escritura. Hoy en día, incluso se pueden usar como ordenadores domésticos (ocio, ofimática, juegos, correo



Figura 4.6. Un ordenador portátil

electrónico, navegación, etc.). También se considera a los *smartphones* (híbridos entre PDA y teléfono móvil) como dispositivos pertenecientes a la misma "familia". La irrupción de este tipo de dispositivos incorporó al usuario avanzado de móviles a este mercado. Vienen equipados con auténticos sistemas operativos tales como Windows CE o Windows Mobile, Palm OS, Symbian OS e incluso Linux (en versiones específicas para estos dispositivos).



Figura 4.7. Dos modelos diferentes de PDA

4.2. ARQUITECTURA

En 1833, Charles Babbage ideó una máquina (mecánica) de cálculo programable mediante tarjetas perforadas: la *máquina analítica*. Esta máquina nunca se terminó de construir, debido a que la tecnología necesaria no estaba lo suficientemente madura. Posteriormente, en 1948, *John von Neumann* propuso utilizar la estructuración ideada por Babbage para construir ordenadores electrónicos. Estos ordenadores disponían de: unidad de control, unidad aritmético-lógica y memoria. A esto le incorporó dos nuevas ideas:

- Representación binaria de la información: los ordenadores van a utilizar el sistema binario para representar datos e instrucciones, idea que ya fue utilizada por Konrad Zuse en 1938 para su Z1.
- Los programas a ejecutar por el ordenador y los datos manejados por éstos debían almacenarse en su memoria de trabajo.

A esta serie de ideas y principios de diseño se le bautizó con el nombre de *arquitectura von Neumann*. En general, los ordenadores basados en la denominada arquitectura von Neumann se componen de los siguientes elementos:

- *Memoria principal*: en la memoria se almacena toda la información que manipula el ordenador, incluyendo los programas que éste ejecuta. Así pues, en la memoria tenemos los programas más los datos que éstos utilizan.
- *Unidad aritmético-lógica*: es el dispositivo encargado de efectuar operaciones aritméticas (como adición, sustracción, etc.) y operaciones lógicas (como OR, NOT, XOR, etc.).
- *Unidad de entrada/salida*: unidad encargada de gestionar la comunicación con dispositivos de lectura/escritura de información, los periféricos (teclado, impresora, pantalla, etc.).
- *Unidad de control*: dispositivo que se encarga de la activación, coordinación y el funcionamiento de los diferentes elementos.

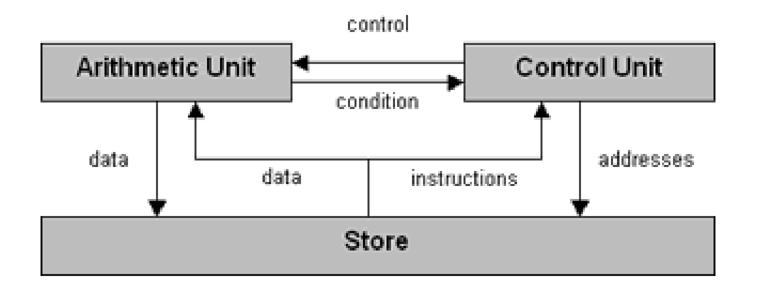


Figura 4.8. Esquema de funcionamiento de la arquitectura von Neumann

La unión de la unidad de control y la unidad aritmético-lógica es, en realidad, la unidad central de proceso, conocida como CPU (siglas que ya conocemos de la sección anterior).

Aunque las tecnologías empleadas en las nuevas computadoras digitales han cambiado mucho desde que aparecieron los primeros modelos de ordenadores en los años cuarenta, la mayoría todavía utiliza las especificaciones de funcionamiento definidas, y los componentes empleados, en la arquitectura von Neumann. Podemos considerar que la arquitectura de una máquina viene determinada por:

- El tipo de procesador que emplea y la clase de dispositivos con los que se conecta.
- La forma en la que se interconectan entre sí los diferentes componentes y la manera en que está hecha la circuitería.
- El modo de funcionamiento.

Actualmente, existen varias arquitecturas: PowerPC, m68k (Mac), Sparc (SUN), etc. Sin embargo, a lo largo de este capítulo vamos a centrarnos en la arquitectura más popular y extendida, no sólo por razones históricas, sino también por su notable rendimiento, su buena relación calidad/precio y el fuerte grado de desarrollo que ha experimentado en los últimos años: la arquitectura PC (de *P*ersonal *C*omputer), también conocida como *x86* (por los sufijos de los "nombres" de los procesadores de Intel que contribuyeron enormemente a popularizarla: 8086, 80286, 80386 y 80486).

Los diferentes componentes y dispositivos que conforman un ordenador se comunican entre sí mediante uno o más buses. Un *bus* no es más que el conjunto de conductores eléctricos (cables, pistas metálicas impresas sobre la placa base del computador) por el cual circulan las señales que corresponden a los datos que comunican dos componentes.

En la Figura 4.9 se muestran dos imágenes con diferentes configuraciones de interconexión de los elementos del ordenador utilizando buses. En la primera imagen todos los elementos comparten el mismo bus, mientras que en la segunda existen dos buses: uno específico para la comunicación entre la memoria y la CPU y otro para la comunicación del resto de periféricos con la CPU. Esta última es la configuración empleada en la actualidad, ya que debe tenerse en cuenta que la mayor parte de las transferencias por los buses se producen entre la memoria y la CPU (la CPU debe acceder a memoria para ejecutar las instrucciones de los programas). La primera, y obvia, mejora para aumentar el rendimiento de los ordenadores consistió en poner un bus específico para las transferencias entre memoria RAM y procesador.

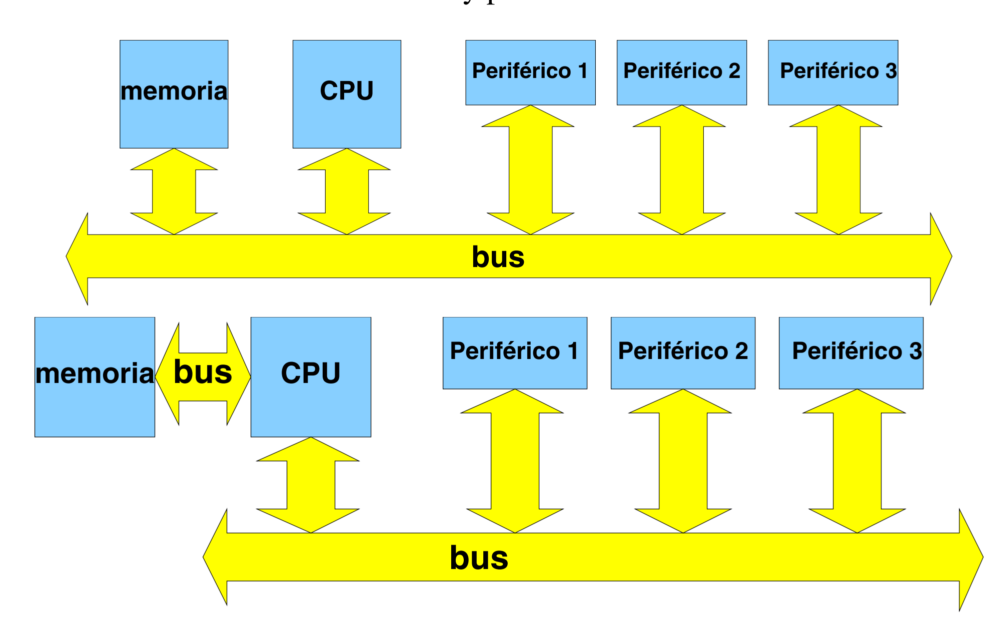


Figura 4.9. Modelo simplificado que muestra el papel de los buses en los ordenadores actuales

Todas las actividades del ordenador se encuentran sincronizadas por un *reloj*. Su frecuencia (o velocidad) se mide en *herzios* (número de ciclos de cómputo que se pueden realizar por segundo). Para que el lector pueda familiarizarse con su empleo, damos el significado de algunas medidas populares.

- Un herzio (Hz) significa una vez (un ciclo) por segundo.

- Un kiloherzio (KHz) significa 1.000 veces por segundo.
- Un megaherzio (MHz) significa $1.000.000$ (un millón) de veces por segundo.
- Un gigaherzio (GHz) significa 1.000 millones de veces por segundo.

Realmente, en un ordenador personal nos encontramos con dos tipos de relojes, aunque en la práctica sólo hay uno, el del bus:

- El reloj externo o de bus. Es el encargado de determinar la velocidad a la que se comunica la CPU con el resto de elementos del ordenador. Su frecuencia es del orden de centenares (incluso miles) de MHz, normalmente 266, 333, 400, 533, 800, 1.000 ó 1.066 MHz (siendo las dos últimas las velocidades de las placas base más recientes en el momento de redactar este capítulo).
- El reloj interno: multiplica la frecuencia del reloj externo (por tanto, realmente es un multiplicador y no podemos decir que sea exactamente un reloj). Se encarga de determinar el ritmo de funcionamiento interno de la CPU, es decir, la velocidad con la que ejecuta instrucciones cuando ya dispone de todos los datos necesarios. Su frecuencia se mide en centenares de MHz o en GHz (actualmente hay microprocesadores que llegan a alcanzar $3{,}8$ GHz).

Sobre la velocidad de reloj

La velocidad de reloj, medida hoy en día en GHz, es un factor muy importante para determinar la prestación de un procesador (es la velocidad a la que realiza las operaciones, es decir, su velocidad de trabajo).

La CPU depende de un cristal de cuarzo para su funcionamiento, que constituye una fuente externa de frecuencia, la frecuencia del reloj.

Ésta se mide en impulsos (también llamados “ticks”) por segundo, descritos como gigaherzios (GHz). Un gigaherzio equivale a mil millones de impulsos por segundo. Por lo tanto, una CPU que funciona a una velocidad de 3 GHz puede realizar aproximadamente unos tres mil millones de operaciones cada segundo.

Podemos decir que el reloj sincroniza la actividad de la CPU. De hecho, el reloj es un generador de señales temporizadas que marcan las fases de ejecución de cada instrucción en la CPU. Hay que tener en cuenta, no obstante, que la ejecución de una instrucción requiere un cierto número de ciclos de reloj (esto depende del procesador).

El rendimiento de un procesador no sólo depende de la velocidad del reloj, sino también de otros factores, como la estructura interna del chip, ancho de palabra o las características del conjunto de instrucciones que ofrece. Por ejemplo, el Pentium Pro podía ejecutar tres operaciones por cada ciclo de reloj, mientras que el Pentium ejecutaba dos por ciclo.

Los dispositivos de cada ordenador son específicos para determinadas arquitecturas (por la forma de conexión, tipos de buses, etc.). Más aún, los programas se preparan

para ser ejecutados en arquitecturas concretas.

Por tanto, si tenemos un tipo de ordenador determinado y queremos ejecutar un programa concreto, deberemos buscar la versión adecuada para nuestro ordenador (arquitectura).

Como hemos visto en el capítulo 2, el sistema operativo (S.O.) se encarga de la gestión de procesos, siendo, por tanto, el responsable de la ejecución de los programas. La ejecución de programas está, pues, íntimamente ligada al S.O. De hecho, el S.O. puede ejecutar los programas porque en los ficheros que los contienen existe información que le indica cómo hacerlo. Esa información es particular para cada S.O. Esto quiere decir que los programas se compilan también para un S.O. concreto. En la práctica, pues, denominaremos *plataforma* de trabajo a la *combinación* de una arquitectura o tipo concreto de ordenador y un S.O. determinado.

4.3. ELEMENTOS DEL COMPUTADOR

En esta sección vamos a tratar de identificar y describir los diferentes elementos básicos que podemos encontrar en un ordenador personal. Tal como muestra la Figura 4.10, podemos apreciar diferentes componentes a primera vista:



Figura 4.10. Un puesto de trabajo informático

- La "caja" , minitorre o unidad central, que contiene el procesador, la memoria, los buses, el gestor de entrada/salida..., es decir, es el computador en sí. Estos componentes se montan sobre la *placa base* (*motherboard*, en inglés). Es un elemento importante, ya que resguarda y sirve de soporte a lo anterior.

  Dentro de la caja tenemos el procesador o CPU, que es el "cerebro" del ordenador, ya que se encarga de controlar al resto de dispositivos y es el que efectúa la mayoría de las operaciones. También nos encontramos con la fuente de alimentación, que es, básicamente, un transformador que se encarga de convertir la corriente alterna que llega a nuestros domicilios en corriente continua necesaria para alimentar a los diferentes dispositivos de nuestro ordenador. Otro elemento importante contenido en la caja, o, mejor dicho, conectado a la placa base que

se sujeta dentro de la caja, es la memoria principal o memoria RAM, donde se guardan los programas en ejecución y los datos que éstos requieren.

En la parte externa de la caja, aparte de ciertas conexiones que encontramos para dispositivos periféricos externos, también tenemos el interruptor de encendido y el botón de reinicio, los cuales permiten controlar el apagado y encendido del ordenador y el *reboot* del sistema[3]. Además, alberga ciertos periféricos que suelen conectarse internamente:

- Discos duros (ocultos en el interior), que sirven para almacenar datos y programas de forma permanente.
- Tarjetas con circuitos integrados, como la de vídeo, responsable de generar las imágenes que se visualizan en el monitor, la de sonido, para poder capturar, procesar y escuchar sonidos (generalmente música), comunicaciones, etc. Todos estos dispositivos, comúnmente denominados "tarjetas", se encuentran, también, en el interior de la caja (igualmente conectadas a la placa base), aunque poseen conectores externos que aparecen en la parte trasera de la misma.
- Unidades de CD-ROM, DVD, disquete, todas ellas accesibles desde la parte frontal. No obstante, queremos hacer notar que las unidades de disquete están prácticamente obsoletas hoy en día y que muy pocos equipos las incluyen actualmente.

Finalmente, tenemos que resaltar el papel que ejercen los *buses* del sistema como elemento vertebrador, ya que sirven de interconexión entre todos los dispositivos mencionados anteriormente. Aunque, propiamente hablando, se encuentran como circuitos integrados en la placa base, podemos considerarlos como elementos internos dentro de la caja.

- Los periféricos externos, que permiten que el ordenador se comunique con el "mundo exterior", tanto a la hora de mostrar o emitir como a la hora de capturar o recibir datos. Podemos destacar:
  - De entrada: teclado, ratón y escáner. Permiten introducir datos en nuestro ordenador (texto, órdenes, imágenes. . .).
  - De salida: monitor, impresora y altavoces. Permiten que el ordenador muestre los resultados del proceso de datos efectuado (ya sea sonido, imágenes o texto).

4.3.1. La caja o unidad central

La caja puede ser un elemento importante aunque no lo parezca. A ella se atornillará la placa base y a ésta se conectarán las tarjetas de expansión. Además, debe tener espacio suficiente para ubicar la fuente de alimentación, los dispositivos de almacenamiento (discos), las unidades de CD/DVD y los ventiladores, todo ello con el espacio

[3]El botón de reinicio genera, normalmente, una interrupción hardware que el sistema operativo "captura", produciendo un reinicio seguro del equipo (equivale a pulsar las teclas Ctrl + Alt + Supr).

y la holgura suficiente para proporcionar una buena ventilación. Por tanto, debe tener ciertas características que la hagan funcional:




Figura 4.11. Dos cajas abiertas: podemos ver el interior del ordenador con la placa base y los dispositivos conectados

- Solidez, es decir, resistencia a golpes o caídas fortuitas. En este sentido, debe estar hecha de un material resistente y ligero, normalmente aluminio o una aleación de dicho material, que proteja los dispositivos internos. Algunas placas llevan incluso paneles laterales dobles.
- Número de ranuras ("huecos") o *slots* suficientes para las unidades (dispositivos de almacenamiento) que se desean colocar.
- Una buena fuente de alimentación y al voltaje adecuado. Un voltaje incorrecto puede dañar otras piezas.
- Fácil acceso para la instalación de dispositivos (extracción fácil de paneles laterales, buena maniobrabilidad, sin necesidad de usar tornillos para la fijación de discos duros. . .).
- Configuración compatible con la placa base que se va a instalar.
- Capacidad de ventilación y refrigeración (número y tipo de ventiladores instalados y existencia de conductos y perforaciones adicionales, bien para dirigir corrientes de aire frío directamente al procesador, bien para disipar la salida de aire de la fuente de alimentación).
- Amortiguación de ruido. Muchas cajas incluyen paneles especiales para absorber el ruido, fuentes de alimentación silenciosas, juntas de goma para aislar el ruido de los dispositivos internos y sistemas antivibración, pero son algo más caras.
- Salida para conectores frontales, tales como USB, FireWire (IEEE 1394), micro o auriculares.

Además, también se pueden valorar aspectos extras, tales como el diseño de la caja (con formas más o menos "exóticas" y/o paneles transparentes), ruedas (con frenos) para poder llevar cómodamente el ordenador de un sitio a otro, etc.

4.3.2. La fuente de alimentación

En electrónica, una fuente de alimentación es un dispositivo o subsistema que convierte la corriente alterna de la red de distribución de la energía eléctrica en otro tipo de corriente eléctrica adecuado para el uso que se le vaya a dar.

En los ordenadores, la fuente de alimentación, también denominada transformador, es una parte esencial, ya que transforma la corriente eléctrica alterna de 220 voltios en corriente continua adecuada para alimentar los diferentes componentes del computador.

En la Figura 4.12 podemos apreciar que de la fuente de alimentación parten cables (de color rojo, amarillo y negro) para alimentar la placa base, el lector de CD-ROM, los discos duros, la disquetera, etc. Los dispositivos que se conectan directamente a la placa, como las tarjetas de expansión y el procesador, se alimentan de la propia corriente que recibe la placa base.



Figura 4.12. Imagen de una fuente de alimentación

4.3.3. La placa base

La placa base constituye la circuitería principal del ordenador y es el lugar donde se encuentran las conexiones básicas para todos sus componentes, los cuales giran en torno al microprocesador. Determina el futuro crecimiento de la capacidad de cualquier ordenador, en función del tipo de buses, número de ranuras para insertar tarjetas, velocidad del bus, etc.

La placa base presenta varias ranuras (*slots*) y zócalos (*sockets*) que permiten insertar fácilmente componentes y dispositivos, algunos *chips* soldados en la placa y *buses*) y *conectores*, los cuales constituyen "pistas" para la transmisión de datos y el control de la circuitería. En la Figura 4.13 se muestran dos imágenes de sendas placas base: en la imagen de la derecha se explicitan mediante rótulos algunos de sus principales elementos.

La placa base constituye el "armazón" o esqueleto principal sobre la cual se montan diferentes dispositivos electrónicos, todos ellos interconectados entre sí a través de los buses y conectores. Los elementos principales que podemos distinguir son:

- El procesador o CPU (también denominado microprocesador o micro a secas): se monta habitualmente sobre un zócalo, aunque existen modelos que se montan sobre una ranura (antiguamente).



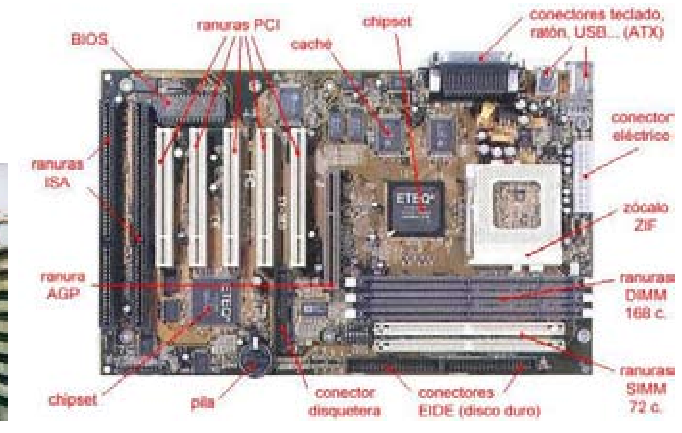


Figura 4.13. Ejemplos de placas base. La imagen de la derecha permite identificar sus componentes básicos

- La memoria principal o memoria RAM (es la memoria necesaria para el trabajo habitual del ordenador): se conecta insertando los módulos correspondientes en unas ranuras específicas.
- Las tarjetas de ampliación o de expansión: se montan sobre unas ranuras de expansión que, al atornillar la placa base a la caja, quedan próximas a la parte posterior de esta última. Estas tarjetas tienen aplicaciones variopintas y, en general, dotan de nuevas funcionalidades y características al ordenador. Así, por ejemplo, encontramos tarjetas para procesar (profesionalmente) sonido o vídeo (como el que podemos capturar con una cámara de mano), para facilitar la interconexión de equipos en red, para poder sintonizar canales de televisión (incluso TDT[4]), etc. Estas ranuras de expansión, en realidad, están conectadas a un bus (o conjunto de buses) en la placa, denominado bus de expansión y que es, habitualmente, el bus PCI (más adelante comentaremos el significado de estas siglas y conoceremos algunas características de este tipo de bus).
- Los chips de control (*chipset*), como, por ejemplo, la BIOS (del inglés *Basic Input-Output System*) y su memoria CMOS (del inglés *Complementary Metal Oxide Semiconductor*). Su misión principal es la de actuar de enlace entre los diversos buses, el procesador y la memoria RAM, principalmente.
- Los conectores para dispositivos internos, tales como discos duros, lectores y grabadores de CD-ROM o DVD±R, o disqueteras, fundamentalmente. Principalmente estamos hablando de buses tipo IDE o, más modernos, SATA (ya comentaremos más adelante algunas cosas al respecto).
- La pila que alimenta la memoria CMOS de la BIOS y un reloj de tiempo real (para recordar la fecha y hora).

En la actualidad, varias placas base llevan integrados, de serie, algunos dispositivos que se consideran "esenciales" y sin los cuales no se vende prácticamente ningún

[4]TDT son las siglas de Televisión Digital Terrestre, el nuevo estándar de televisión (comunicación audiovisual) que se está implantando en nuestro país y que concluirá con el denominado "apagón analógico", que tendrá lugar dentro de poco.

ordenador. Es el caso de la tarjeta gráfica (necesaria para que el ordenador pueda mostrar imágenes en el monitor), un dispositivo procesador de la señal acústica (tarjeta de sonido) y un módem o tarjeta Ethernet (necesario para poder conectar nuestro equipo a una red local o a la red corporativa de una empresa de comunicaciones o de una universidad). Puesto que a casi cualquier ordenador que se compra se le exige un mínimo de capacidad multimedia (por lo menos la capacidad de procesar imágenes y reproducir vídeos y música) y la posibilidad de conectarse a Internet, es necesario adquirir estos dispositivos si no vienen en la propia placa base.

Obviamente, el precio de las placas que incluyen estos dispositivos suele ser ligeramente superior al de las que no los incluyen. Sin embargo, esto también va a depender de cuestiones tales como la "calidad" de la placa (y en esto influye mucho el fabricante) y la calidad de los dispositivos integrados. En principio, podemos asumir que, normalmente, adquirir estos dispositivos por separado puede ser algo más costoso, aunque esto no tiene porqué ser siempre así. También es cierto que los dispositivos integrados suelen ofrecer menos prestaciones que muchos de los que se venden por separado. Además, el hecho de adquirir tarjetas por separado nos permite poder elegir entre diversas gamas (con prestaciones medias o altas) y buscar en el mercado las características que específicamente nos interesen. Un ejemplo servirá para ilustrar esta cuestión: es difícil encontrar tarjetas de sonido integradas en placa que ofrezcan las características que requiere un ingeniero de sonido.

¿Sabías que…?

Desde hace ya algunos años, incluso aunque trabajemos en "modo texto" (consola) como permiten algunos sistemas operativos (Linux por ejemplo), la tarjeta gráfica es necesaria para enviar la señal correspondiente al monitor. Incluso si la tarjeta gráfica no funciona o no está correctamente conectada, el ordenador podría no arrancar al encenderlo.




Figura 4.14. Otra imagen de una placa base. La imagen de la derecha es la misma placa pero con varios dispositivos conectados (procesador, memoria…)

En los siguientes apartados, y también en la sección 4.4, daremos más detalles sobre el funcionamiento y características básicas de varios de estos elementos. Tan

sólo nos resta comentar en qué tenemos que fijarnos a la hora de optar por una placa base u otra:

- El *chipset*, es decir, el conjunto de chips que controlan la placa, es un aspecto importante en la elección. El siguiente apartado está dedicado concretamente a comentar esta cuestión.
- El tipo de zócalo en el que se instala el microprocesador es un factor que hay que conocer, aunque sólo en el caso en el que decidamos comprar nuestro ordenador por piezas y montarlo nosotros mismos. Obviamente, los ordenadores que compramos (sean de marca o no) vienen con procesadores compatibles con la placa base.
- El tipo de buses que incorpora la placa. Los buses son los canales para la transmisión de datos entre microprocesador, memoria y dispositivos, por lo que determinan el tipo de dispositivos que podremos conectar a nuestro ordenador. Normalmente éste no es un factor relevante, ya que todas las placas modernas cubren las necesidades básicas a este respecto (PCI, AGP, IDE, USB 2.0). Sin embargo, sí puede ser un factor decisorio en el caso en el que queramos conectar a la placa algún dispositivo o tarjeta antigua (que requiera bus ISA, por ejemplo), o bien busquemos ciertas características más avanzadas que permitan "ampliar" las capacidades del equipo informático a corto plazo (por ejemplo, queremos una placa que incorpore bus FireWire o PCI Express). De los buses hablaremos en el apartado 4.3.11.
- El tipo y número de ranuras de expansión que tiene la placa es un aspecto estrechamente relacionado con el anterior. Tan sólo queremos apuntar que el número de ranuras determina, ya sea directa o proporcionalmente, el número máximo de dispositivos que podemos conectar simultáneamente a nuestro ordenador.
- El tipo y número de ranuras para la memoria RAM tampoco es un factor por el que haya que preocuparse normalmente: las placas base recientes cubren sobradamente nuestras necesidades básicas a este respecto. Este factor va a determinar la capacidad de ampliación de memoria de nuestro equipo. Básicamente, hemos de comprobar que la placa esté al tanto del "estado del arte" en lo que se refiere a las tecnologías de empaquetado de circuitos de memoria y en lo que concierne al sistema de transferencia y acceso de datos.
- Las frecuencias de reloj externo soportadas constituyen una característica importante de la placa. Como hemos visto en la sección anterior, el reloj de la placa (reloj externo) marca la velocidad a la que se va a comunicar el procesador con el resto de elementos conectados a la placa (nos interesa que sea lo más rápida posible para evitar en la medida de lo posible los ciclos de espera ociosos de la CPU); pero también determina indirectamente la frecuencia de ciclo del procesador, es decir, la velocidad a la que va a operar éste. Por tanto, es un factor importante en el que hay que fijarse para calibrar la potencia de nuestro ordenador.
- El fabricante es, finalmente, otro aspecto a considerar. En realidad, estamos hablando de la "marca" de la placa. Hay una serie de fabricantes en el mercado que

ofrecen mayores garantías que otros: calidad de materiales, fiabilidad y buenas prestaciones. Normalmente, sus placas son más caras; aunque hay algunos que tienen precios económicos y, por consiguiente, presentan una excelente relación calidad/precio. Algunos de los fabricantes de mayor prestigio son Intel, Asus, MSI, Gigabyte, Abit, Tyan...

¿De marca o clónico?

Desde hace ya varios años se suele optar, a la hora de comprar un ordenador personal, por adquirir equipos montados en tiendas especializadas en informática. Estas tiendas suelen ofrecer ofertas de equipos con buena calidad, aprovechando compras a mayoristas, excesos de *stock*, etc. A estos ordenadores se les denomina comúnmente "clónicos".

Los equipos se montan en la propia tienda, a partir de piezas adquiridas por separado y compatibles entre sí. Se ofrecen equipos que abarcan una amplia gama de precios, en función de la calidad de los componentes empleados en el montaje. La tienda aplica un margen de beneficios en concepto de montaje y/o asistencia técnica. Podemos revisar la lista de componentes empleados (marca y características) antes de elegir el equipo que queremos comprar.

También podemos elaborar nosotros una lista y pedirle a la tienda que compre las piezas y monte el equipo en base a un presupuesto determinado. O bien podemos montar nosotros el equipo, pero en ese caso tendremos que cuidar de todos los detalles personalmente. En la práctica, con una buena selección de componentes, y si se conoce bien el "mercado", podemos conseguir ordenadores con una excelente relación precio/calidad. Y contando con el valor añadido de comprar estrictamente lo que necesitamos (o lo que prevemos que podemos necesitar a corto o medio plazo).

Igualmente, podemos desglosar el presupuesto a nuestro antojo (por ejemplo, podemos decidir comprar una tarjeta gráfica de gama alta y gastarnos más dinero en este epígrafe a costa de adquirir una tarjeta de sonido más modesta y con menores prestaciones).

Otra ventaja más: normalmente, podemos elegir si queremos un sistema operativo preinstalado o no. Si estamos pensando en trabajar sólo con Linux, nos podemos ahorrar la bonita cantidad que cuesta la licencia de Windows. Incluso algunas tiendas venden clónicos con un sistema Linux preinstalado y hardware 100 % compatible.

Por el contrario, los ordenadores de marca vienen con una configuración de fábrica que no puede alterarse. Normalmente, los componentes son adecuados y con una calidad proporcional al precio de venta del equipo. En este sentido, nos ahorran "faena", pero como contrapartida tenemos que "tragar" con los dispositivos que vienen de serie (igual nos interesa una tarjeta de sonido profesional y nos da igual la calidad de la tarjeta gráfica, y sucede al revés).

continúa en la página siguiente ...

. . . viene de la página anterior

Los equipos de marca son, igualmente, ordenadores personales, por lo que podemos abrirlos y montar nuevas tarjetas y/o dispositivos que adquiramos de manera independiente. Lamentablemente, muchos de estos equipos pierden la garantía si son abiertos por personal no autorizado (y suelen llevar un sello que se rompe al abrirlos para poder demostrarlo), lo que los convierte en la práctica en "no configurables" (al menos mientras dure el periodo de garantía), perdiéndose así uno de los mayores atractivos de los ordenadores personales.

También ocurre que suelen venir con un sistema operativo preinstalado, generalmente de la familia Windows, cuyo coste encarece el precio del equipo (no digamos si, además, se incluye la instalación de un paquete ofimático tal como Office de Microsoft). Siempre podemos no aceptar la licencia y pedir que nos reembolsen el precio del software no deseado, pero éste suele ser un proceso farragoso que suele costar tiempo y para el que se ha de ser muy tenaz y paciente.

Otro inconveniente suele ser el precio. Normalmente, los clónicos equiparables a los de marca eran bastante más baratos (casi la mitad de precio). Pero esto no es así en la actualidad, ya que los ordenadores de marca han bajado más el precio. Por otro lado, también han aumentado su "flexibilidad": se ofrece una gama más amplia de equipos a elegir con diferentes prestaciones, marcas y modelos de procesador a precios bajos, medios y altos. En este sentido, ya no existen tan grandes diferencias en precio y configuración como para decantarse claramente por un clónico.

Las ventajas que ofrecen los ordenadores de marca tienen que ver con un mayor control de calidad (los controles de calidad de empresas como Dell, HP, etc., son exhaustivos), instrucciones de uso e instalación que acompaña al producto, manuales traducidos (aunque algunas traducciones son de una calidad muy pobre) y servicios "especiales": transporte e instalación en casa, garantía extendida (tres años), servicio técnico, asistencia telefónica, etc.

4.3.4. El *chipset*

El *chipset* es el conjunto de chips que, montados sobre la placa base, controlan ciertas funciones del ordenador y el modo en que interacciona el microprocesador con la memoria y demás elementos. En realidad, podemos considerar que, en cierta forma, todos los elementos que hemos comentado en el apartado anterior están "unidos" mediante este conjunto de chips específicos. Las principales funciones que desempeña esta circuitería se podrían sintetizar en las siguientes:

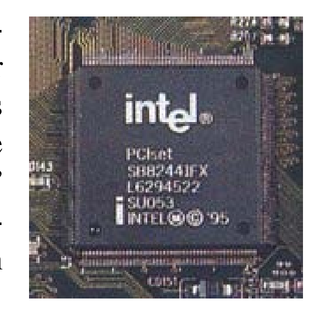


Figura 4.15. Un chipset de Intel

- Control de las transferencias de datos en los buses.
- Puesta en marcha de las señales que activan el traspaso de datos entre los diferentes buses.
- Control de las transferencias de datos a o desde la memoria principal (RAM).

- Control de algunos periféricos, como, por ejemplo, algunos discos duros.

Por la relevancia y lo esencial de las funciones que controla, se puede afirmar que de la calidad del chipset depende en buena medida la eficiencia del ordenador. En este sentido, debemos tener en cuenta que el chipset y la placa base "van de la mano" y no podemos adquirirlos por separado. Eso sí, ésta es una de las características de la placa en la que podemos fijarnos a la hora de perfilar nuestra elección. Tengamos en cuenta, igualmente, que no hay una correspondencia exacta entre fabricantes de chipsets y de placas base, ya que diferentes placas base de fabricantes distintos pueden montar los mismos chipsets. Para que nos hagamos una idea, he aquí una breve lista de los fabricantes más conocidos de chipsets: Intel, AMD, VIA, ALI, SiS, Opti...

4.3.5. BIOS: *Basic Input-Output System*



Figura 4.16. La imagen de una BIOS

Normalmente, cuando se le pregunta a cualquier persona que, sin ser informático, tenga unos mínimos conocimientos de informática qué es lo que ocurre cuando encendemos el ordenador (lo que se conoce como "arranque" del mismo), ésta nos responderá que se ejecuta el sistema operativo, el cual toma el control de los dispositivos y nos permite trabajar de manera fácil e intuitiva con el resto de programas que hay instalados.

Hasta cierto punto, esto es cierto. ¿Y hasta dónde? Pues hasta el punto en que *realmente* se ejecuta el sistema operativo, lo cual no coincide, ni mucho menos, con el momento en el que encendemos nuestro ordenador. Lo primero que hace el ordenador cuando arranca es ejecutar una serie de instrucciones que están grabadas en un chip de memoria de la placa base. A este chip se le denomina BIOS, que podríamos traducir como sistema básico de entrada/salida (del inglés *Basic Input-Output System*). Después, es la BIOS la que selecciona el dispositivo desde el cual se va a intentar cargar un sistema operativo (normalmente el disco duro). Para ello, como último "acto de servicio", la BIOS carga en memoria el contenido del *sector de arranque* (o sector cero) del dispositivo elegido. Allí debería haber un pequeño programa (no hay mucho espacio que digamos en dicho sector de arranque), el cual se denomina cargador[5], y cuya misión consiste fundamentalmente en cargar en memoria RAM el contenido del fichero (imagen) con las instrucciones del sistema operativo (el ejecutable, vamos). Es en este preciso momento cuando realmente entra en ejecución el sistema operativo y toma el control de los dispositivos y del ordenador.

[5]Dos ejemplos de "cargadores": LILO, por *LInux LOader*, y GRUB (*GRand Unified Bootloader*) capaces de arrancar las imágenes en disco de varios sistemas operativos (Linux, Windows XP, Windows Vista, etc.); aunque el segundo es más potente que el primero.

Antiguamente, la BIOS era una memoria de tipo ROM (memoria de sólo lectura; del inglés *Read-Only Memory*): PROM (*Programmable Read-Only Memory*) o EPROM (*Erasable Programmable Read-Only Memory*); pero hoy día, normalmente, es de tipo Flash. La memoria Flash es una memoria no volátil que puede ser borrada y reprogramada eléctricamente, por lo que se puede modificar su contenido cuando se requiere actualizarlo. De hecho, los fabricantes de BIOS suelen proporcionar *software* para poder actualizar la programación grabada en ella. Algunos de los principales fabricantes de BIOS son *American Megatrends* (AMI), *Award Software International* y *Phoenix Technologies*. Precisamente, las dos últimas empresas se fusionaron en 1998.

Vamos a estudiar con algo más de detalle en qué consiste la secuencia de arranque desde el momento en el que se ejecuta el programa almacenado en la BIOS. Esta secuencia consta, en realidad, de varios pasos en los cuales la BIOS efectúa estas acciones:

- Comprobar qué hardware hay conectado al equipo (buses, placa, etc.).
- Realizar algunas pruebas para comprobar que todo funciona correctamente (chequeo de memoria, teclado y tarjeta gráfica presentes. . .).
- Asignar recursos a las tarjetas y dispositivos[6] para que puedan comunicarse con y ser manipulados por el sistema operativo. Estos recursos consisten, básicamente, en asociar un número identificador de IRQ (petición de interrupción; del inglés *Interrupt ReQuest*) y un puerto de entrada/salida (podemos verlo como una dirección de memoria necesaria para la comunicación entre el sistema y la tarjeta o dispositivo). En el apartado 4.4.2 comentaremos más cosas al respecto.
- Determinar la unidad o dispositivo desde el cual se va a proceder al arranque del sistema operativo (con frecuencia el disco duro; aunque también se puede arrancar desde un DVD o CD-ROM, sobre todo si estamos realizando la instalación de un sistema operativo) y cargar en memoria para su ejecución el programa que haya almacenado en su sector de arranque, tal como dijimos con anterioridad.

¿Qué son las interrupciones?

Los dispositivos de nuestro ordenador necesitan de algún mecanismo que les permita "llamar la atención" del sistema operativo. Pensemos, por ejemplo, en la siguiente situación: la tarjeta *ethernet* (dispositivo que permite que nuestro ordenador se comunique con otros equipos en una red de ordenadores) de nuestro PC tiene datos que el sistema operativo debería "pasar" al programa de transferencia de ficheros que los solicitó.

Por otro lado, puede ocurrir que ejecutemos un programa reproductor de música para escuchar una serie de ficheros MP3. Dicho programa le envía datos (sonidos) al sistema operativo para que "salgan" por la salida de audio (altavoces) de la tarjeta de sonido de nuestro ordenador.

continúa en la página siguiente . . .

[6] También se asignan recursos a los circuitos controladores de ciertos buses, tales como USB.

. . . viene de la página anterior

De alguna forma, el sistema operativo necesita decirle a la tarjeta dónde se encuentran (en qué posición de memoria) esos datos.

Precisamente, para todo esto se necesitan las interrupciones y los puertos de entrada/salida. El mecanismo de interrupciones permite enviar señales hardware (de entre una batería que vienen predeterminadas en la circuitería del ordenador) que el sistema operativo está preparado para "capturar". Cada una de estas interrupciones está asociada a un dispositivo concreto. De esta forma, cuando un dispositivo "llama al timbre", es decir, emite la señal correspondiente a su interrupción, el sistema operativo ya sabe quién está llamando.

Los puertos de entrada/salida permiten pasar información complementaria en la comunicación: por ejemplo, para que el sistema indique al dispositivo la dirección de memoria donde están los datos que debe procesar.

¿Qué ocurre si a dos dispositivos distintos se le asigna la misma interrupción? En algunas ocasiones, no pasará nada, ya que hay algunos dispositivos que están preparados para compartir una misma interrupción y, de alguna forma, la situación es gestionada correctamente por los dispositivos implicados. Pero otras veces tendremos un conflicto: uno de los dos dispositivos implicados no funcionará (o no lo hará correctamente), ya que está "tapado" por el otro. La solución puede ser difícil, aunque lo más cómodo en estos casos, si se dispone de ranuras libres en el bus PCI de la placa, es cambiar a otra ranura uno de dichos positivos. Al arrancar, se reasignan las interrupciones y, normalmente, el problema se resolverá solo.

Tal como hemos apuntado con anterioridad, la memoria de la BIOS es especial, ya que es capaz de almacenar información de manera permanente sin necesidad de alimentación continua. Debes saber que es posible actualizar su contenido (el programa que tiene almacenado) mediante aplicaciones especiales que proporciona el fabricante. De hecho, en algunas ocasiones esto resulta incluso conveniente: el programa original puede tener *bugs* (errores de programación), pueden existir problemas con ciertos dispositivos, etc. Si quieres averiguar más cosas al respecto, te sugerimos que visites la siguiente página web: `http://www.wimsbios.com/`.

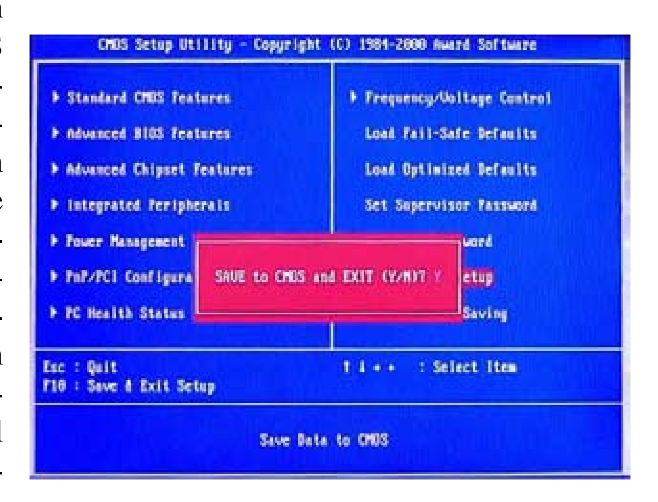


Figura 4.17. Menú de configuración de una BIOS

El programa almacenado en la memoria de la BIOS, por otro lado, es capaz de manejar diferentes opciones de configuración, relacionadas, sobre todo, con el modo de funcionamiento de la placa base y con la configuración de los dispositivos que a ella se conectan. Debes tener presente que algunas características del hardware son configurables; por ejemplo, establecer la velocidad de la CPU dentro de un cierto rango,

configurar la velocidad del bus...

Todas estas opciones, y más (relacionadas con el propio funcionamiento de la BIOS, como, por ejemplo, establecer el orden en que se ha de buscar dispositivos de arranque al encender el ordenador), se pueden configurar a través de un programa que está disponible por un breve intervalo de tiempo (segundos, habitualmente) durante la fase inicial de comprobaciones de la BIOS al encender el ordenador. Con frecuencia, se puede acceder a ejecutar este programa pulsando la tecla Supr, F2 o bien ESC. En el arranque aparecen, brevemente, ciertos mensajes de la BIOS y en uno de ellos se suele indicar qué tecla hay que pulsar para acceder a dicho programa. Si esto no es así, habremos de consultar el manual de la placa que nos deben haber entregado con la compra del ordenador.

La Figura 4.17 muestra un ejemplo de programa de BIOS. Nótese que las opciones, la disposición y el aspecto de los menús del programa pueden cambiar según el programa que haya almacenado en la BIOS. Esto dependerá de la marca de la misma. En el ejemplo de la figura se trata de un programa de BIOS elaborado por la empresa *Award Software*, actualmente fusionada con *Phoenix Technologies*, y dedicada, entre otras actividades, a la fabricación de BIOS y programas para BIOS. Tan solo nos resta por comentar que los datos con los valores de las opciones de configuración que podemos modificar con este tipo de programas se guardan en una memoria especial de tipo *CMOS* (del inglés *Complementary Metal Oxide Semiconductor*), cuyos contenidos no se borran al apagar el equipo, ya que sus circuitos se mantienen alimentados con la pila de la placa.

4.3.6. El microprocesador

El microprocesador junto con la placa base constituye la parte más importante del ordenador. Se puede afirmar que el procesador es el "cerebro" del ordenador, ya que se encarga de realizar las operaciones y de dirigir la interacción con los diversos dispositivos. El procesador es un chip complejísimo, ya que cuenta con millones de elementos y circuitos integrados, tal como puede apreciarse en la Figura 4.18. Todo lo que le rodea y que se conecta bien de forma directa, bien de forma indirecta, no son más que dispositivos, mediante los cuales el microprocesador interactúa con los usuarios.

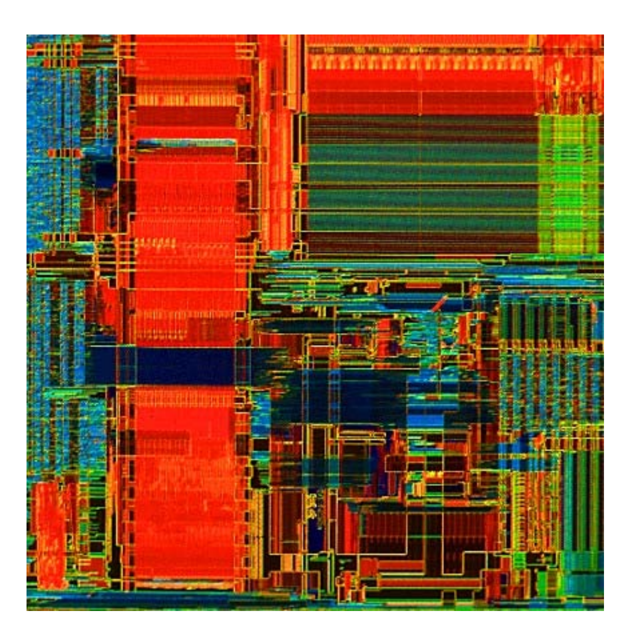
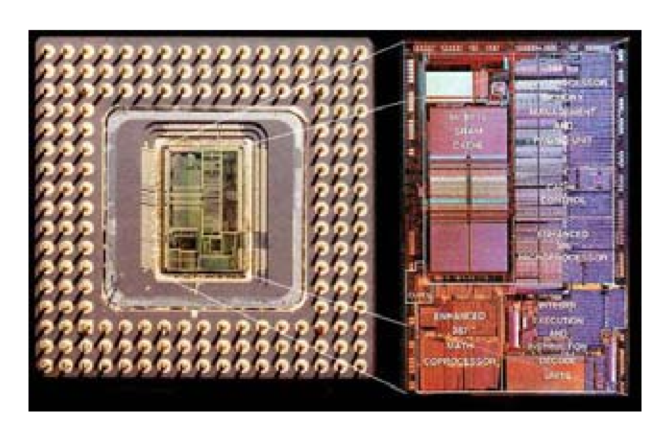


Figura 4.18. Los circuitos de un procesador Intel Pentium

El microprocesador suele denominarse procesador, a secas, aunque se le conoce por el término CPU (que ya vimos que significa *Central Processing Unit*, o sea Unidad Central de Proceso). A lo largo de este capítulo emplearemos in-

distintamente ambos términos para referirnos a él. La CPU se compone de la Unidad Aritmético-Lógica (UAL) y de la Unidad de Control (UC). La primera se encarga de realizar toda la operatoria, mientras que la UC se encarga de interpretar las diversas instrucciones, tomar operandos, escribir/leer en memoria, mover datos a/desde los registros. . . Esta última trabaja en base a un *reloj maestro*. que coordina la ejecución de todas las operaciones que realiza el procesador.



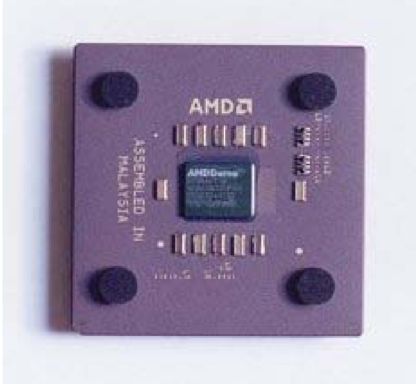


Figura 4.19. Imágenes de dos procesadores: a la izquierda un 486 de Intel, a la derecha un Duron de AMD

Además, dentro del encapsulado del procesador se encuentran otros circuitos importantes, como son la memoria caché y el coprocesador matemático, encargado específicamente de realizar las operaciones con números de punto flotante (más costosas que las correspondientes con números enteros, de ahí lo de dedicarle un procesador específico). Los modelos más recientes de procesador incluyen circuitería adicional con más memoria caché.





Figura 4.20. Otros dos procesadores con circuitos adicionales de memoria caché

En cuanto al reloj maestro, hemos de indicar que es una parte principal, ya que determina el ritmo de trabajo del procesador y, por ende, del ordenador. La unidad fundamental de trabajo de este reloj es la cantidad de instrucciones que el microprocesador puede ejecutar en un segundo. Esto viene determinado por la frecuencia del reloj, la cual se mide actualmente en GHz, esto es, el número de ciclos de reloj por segundo. Así, en teoría, un procesador a 1.800 GHz podría ejecutar 1.800 millones de instrucciones por segundo. Lo que ocurre realmente, es que muchas de las instrucciones de los procesadores actuales son ciertamente complejas y requieren de varios ciclos de reloj para su compleción. Además, también influyen otros factores, como veremos

posteriormente. *Grosso modo*, distinguiremos las siguientes partes en el diseño de un procesador:

- El microprocesador propiamente dicho, que es donde se ejecutan las instrucciones. Se compone, como hemos visto, de la unidad de control, la unidad aritmético-lógica, el coprocesador matemático, etc.
- El encapsulado que rodea al microprocesador. Es como una "cubierta" que lo protege de la oxidación y que contiene los cables y el conector para el zócalo.
- La memoria caché, que es una memoria ultra-rápida y de poca capacidad (comparada con la memoria RAM, por ejemplo), sirve fundamentalmente para almacenar las porciones de la memoria principal que, con mayor probabilidad, se van a usar en el futuro. Su uso acelera sustancialmente la velocidad efectiva del microprocesador, ya que actúa como almacén, tanto de instrucciones como de datos recientemente usados o accedidos. Cada vez que el procesador "encuentra" alguna instrucción o dato en la caché, se ahorra el tiempo de acceso a la memoria RAM, ya que el acceso a la caché es mucho más rápido.

Precisamente, el tamaño de la memoria caché es un factor importante a la hora de evaluar las prestaciones y el rendimiento de un procesador, ya que si bien es cierto que hay una relación entre mayor frecuencia del reloj y mayor velocidad efectiva del procesador (a mayor frecuencia, mayor velocidad y, por tanto, rendimiento), has de tener en cuenta que esta última depende, en general, de muchos factores. Algunos de los factores que, por ejemplo, debes considerar son los siguientes:

Arquitectura: es importante tener en cuenta cuestiones relativas al diseño y arquitectura interna del procesador para poderse hacer una idea de las prestaciones del mismo. Nos estamos refiriendo fundamentalmente a la tecnología empleada. Por ejemplo, ¿se trata de un procesador con dos (tecnología Core 2 Duo) o con cuatro núcleos (tecnología Quad-Core de Intel)?, ¿cómo se interconectan entre sí los diversos componentes del procesador?, ¿con qué circuitos se han construido?; en general, ¿es capaz el procesador de efectuar varias operaciones simultáneamente y, por tanto, sacarle más rendimiento a cada ciclo de reloj acercándose al máximo utópico de una instrucción ejecutada por ciclo?

En esto último influyen factores tan variopintos como el número de niveles en la *pipeline* (circuitos encargados de la paralelización en la ejecución de las diversas fases en las que se descomponen las instrucciones del procesador), la capacidad de multiescalabilidad del procesador o el empleo de tecnologías como el *hyperthreading* (en la gama de procesadores Pentium IV de Intel) o el Dual Core (Intel) o doble núcleo (AMD). También tiene una gran influencia en las prestaciones del procesador si éste es capaz de manipular 64 ó 32 bits de una "tacada" (en sus operaciones de transferencia a memoria, por ejemplo). Para "ser" de 64 bits, el procesador debe contar con registros de 64 bits y también con instrucciones que puedan manipular datos con esta anchura. Precisamente, se define el *ancho de palabra* de un procesador como la cantidad de bits que es capaz de procesar simultáneamente (básicamente, el tamaño de sus registros internos).

Para que el lector pueda hacerse una idea de lo que estamos hablando, explicaremos en qué se basa la técnica de *pipeline*. Ésta es una técnica de ejecución de las instrucciones consistente en dividirlas en distintas fases, de manera similar a las cadenas de fabricación en serie. Así, se puede determinar si dos fases no tienen ninguna dependencia entre sí. En ese caso se pueden ejecutar *en paralelo*, cada una en un nivel distinto de la *pipeline*, esto es, en una línea distinta de "montaje", siguiendo con el símil de la fabricación en serie.

La arquitectura del procesador depende del fabricante y del modelo. Hay dos arquitecturas populares en las que se basan los procesadores fabricados para ordenadores personales: *x86* (usada por el $95\,\%$ de los ordenadores personales) y *PowerPC* o *PowerMac* (usada hasta principios de 2006 por los ordenadores de la familia Macintosh). A igual frecuencia de reloj, los PowerPC eran más rápidos (aunque también algo más caros). Tras el paso dado por Apple con la adopción de procesadores de la arquitectura x86[7] (Intel Xeon Dual Core), prácticamente la totalidad del mercado de ordenadores personales emplea procesadores de la familia x86.

Debes saber que, en realidad, x86 es la denominación genérica dada a ciertos procesadores fabricados por Intel, sus "compatibles" (procesadores que entienden su juego de instrucciones básico y, posiblemente, añaden otras nuevas) y a la arquitectura básica de estos procesadores, debido a la terminación de sus nombres: 8086, 80286, 80386 y 80486. Los sucesores de este último presentaban muchos avances en la arquitectura, aunque seguían siendo procesadores de 32 bits y mantenían compatibilidad hacia atrás con todos ellos: fueron los Pentium. En realidad, el Pentium ya no era un procesador de la familia x86, pero debido a su compatibilidad con estos procesadores y al hecho de ser fabricado por la misma empresa (Intel), se le consideró como un miembro más del grupo x86, y ya por extensión, se consideraron igualmente a los modelos y versiones posteriores de Pentium (incluso los de 64 bits, mixtos). En la actualidad, existen dos sucesores de 64 bits para esta arquitectura:

- IA64 (Intel Architecture-64), empleada en los procesadores Itanium de Intel y *no* compatible con x86, excepto bajo emulación.
- AMD64, también conocida como x86-64 (denominada así puesto que es compatible con la "vieja" familia x86 de 32 bits), que es, básicamente, una extensión de 64 bits de la familia x86, ya que duplica el juego de instrucciones de 32 bits (y añade otras nuevas instrucciones exclusivamente de 64 bits). Para que los programas puedan sacar partido de esta arquitectura deben ser recompilados a 64 bits (es decir, recompilados para generar código máquina que emplee el nuevo juego de instrucciones de 64 bits).

Hay dos fabricantes principales de microprocesadores x86: *Intel* y *AMD*. A igual frecuencia de reloj, los AMD son más rápidos en algunas operaciones, y más lentos en otras, por lo que no puede decirse que haya una superioridad tecnológica clara; al menos hasta la aparición de la tecnología de doble núcleo. Dejando de

[7] A finales de 2006, la compañía había migrado completamente de la arquitectura PowerPC a x86.

lado, de momento, esta tecnología, podemos decir que la competencia se estableció, por un lado, con Intel, aumentando la frecuencia de reloj y, por otro, con AMD, haciendo más cosas en cada ciclo de reloj. Es por ello, que los procesadores Athlon de AMD a 2 GHz ofrecían prestaciones similares a los Pentium IV a 3 y 3,2 GHz.

Sin embargo, la aparición de la tecnología Dual Core de Intel ha supuesto un nuevo e importante avance en el desarrollo de procesadores. La idea consiste en tener dos núcleos de proceso en el mismo encapsulado. Es casi como tener dos procesadores independientes, con lo cual se aumenta la paralelización y se incrementa en gran medida el grado de aprovechamiento de cada ciclo de reloj. Además, estos procesadores incorporan todos los desarrollos tecnológicos realizados en los últimos tiempos y vienen equipados con más memoria caché y de mayor tamaño. De momento, AMD ha respondido al desafío con su propia gama de procesadores de doble núcleo; pero Intel (el desarrollador original de la tecnología, no olvidemos) ya va a sacar al mercado los *Quad-Core*, procesadores con cuatro núcleos, y anuncia, para el 2010 ó 2012, verse capaz de sacar procesadores ¡con hasta 32 núcleos de proceso!

Memoria caché: los circuitos de memoria caché y su tamaño, son otro elemento que contribuye notoriamente a mejorar las prestaciones de un procesador. En general, podemos afirmar que a mayor cantidad de caché, mayor eficiencia en ciertas aplicaciones (ofimática, multimedia. . .). Resumiendo, podemos distinguir dos tipos de memoria caché en el procesador: la caché *L1* (o de nivel 1) y la caché *L2* (o de nivel 2). La L1 es ultra-rápida y pequeña (su tamaño se mide en centenares de KB) y forma parte del propio microprocesador; es decir, que se dispone en chips integrados dentro de su propio encapsulado. La L2 es algo más lenta (no suele formar parte del encapsulado y, por ello, el tiempo de acceso es algo mayor) y de mayor tamaño (se mide en MB; normalmente 1, 2 ó 4 MB). La caché de nivel 2 se dispone en circuitos adicionales alrededor del encapsulado. Además, se pueden disponer circuitos con memorias caché adicionales (algunas en la placa base). Éste es el caso de algunos procesadores que utilizan memorias de caché de nivel 3 (L3) y con ello consiguen mejorar sus prestaciones (sobre todo con aplicaciones multimedia). Las memorias de caché de nivel 3 pueden llegar a tener centenares de MB.

Frecuencia del bus: ya sabemos que la CPU se relaciona con el resto de componentes a través de uno o más buses que se hallan integrados en la placa (son circuitos de la misma). En muchas ocasiones, ocurre que el procesador envía datos, o bien una petición por el bus, y no puede volver a actuar hasta que se haya completado el envío de datos, o bien la petición. En esas situaciones, el procesador permanece ocioso (no efectúa ninguna operación) y se desperdician ciclos de reloj. Por tanto, de poco sirve que el procesador sea muy rápido si el bus es muy lento, ya que puede estar gran parte del tiempo esperando que lleguen los datos con los que trabajar. Hay que tener presente que no sólo interesa aprovechar al máximo cada ciclo de reloj, sino también minimizar, en la medida de lo posible, el número de ciclos ociosos de espera: ambos factores afectan directamente al ren-

dimiento del procesador. A cuanta mayor frecuencia opere el bus de la placa, a mayor velocidad se realizará la transferencia de datos y, por tanto, existirán mayores posibilidades de reducir los ciclos ociosos de espera en el procesador.

A modo de resumen, queremos mostrar en el Cuadro 4.1 los nombres de algunos de los procesadores más conocidos de la familia x86, junto con una frecuencia de reloj típica[8] y el año de aparición.

Marca	Frec. reloj típica	Año aparición	Característica destacada
8088	4,77 MHz	1981	versión "reducida" del 8086
8086	8 MHz	1983	usado en un PC mejorado
80286	12 MHz	1986	usado en el PC AT
80386	33 MHz	1987	modelo de memoria mejorado, arquitectura de 32 bits
80486	50 MHz	1991	integra caché y coprocesador
Pentium	100 MHz	1993	ejecuta instrucciones en paralelo
Pentium MMX	200 MHz	1997	instrucciones multimedia
Pentium II	450 MHz	1997	caché de nivel 2 fuera del chip
Pentium III	1 GHz	2000	más instrucciones multimedia
Pentium 4	3 GHz	2000	bus más rápido, hiperpipeline, más caché
Itanium	1,4 GHz	2002	arquitectura de 64 bits

Cuadro 4.1. Algunos procesadores x86

Ahora, vamos a comentar algunas de las características más destacadas de las gamas de procesadores de la familia x86 que podemos encontrar en los ordenadores personales desde hace unos cuatro años hasta la actualidad, considerando tanto los denominados ordenadores de sobremesa (*desktop computers*) como los portátiles. En lo que se refiere al fabricante Intel, podemos destacar:

- *Celeron*: son procesadores de la gama Pentium pero con poca caché, sobre todo de nivel 2. Constituyen la gama económica, ya que son los más baratos, pero, obviamente, ofrecen menos prestaciones. Se consideran procesadores de gama media o baja, resultando muy adecuados (en su relación calidad/precio) para tareas de ofimática y contabilidad. También se emplean en las gamas bajas de portátiles, más económicos y con menor consumo, aunque más limitados en cuanto a potencia y rendimiento. Los Celeron se ofertan en varios modelos (muchos ya obsoletos), con frecuencias de reloj que oscilan entre los 950 MHz y los 2,80 GHz. Actualmente, sólo se fabrica la gama denominada *Celeron D*, tienen algo más de caché (hasta el doble en los últimos modelos) y llegan hasta los 3,20 GHz.

- *Pentium 4*: constituyen, en puridad, la última serie de procesadores de la familia x86 de Intel. La mayoría de versiones de este procesador ya están obsoletas y no se fabrican. Los Pentium 4 de última generación emplean la tecnología

[8] Se sacaron diversas versiones de cada uno de estos modelos, que funcionaban con diferentes frecuencias de reloj.

hyperthreading, que se considera pionera de la tecnología de doble núcleo desarrollada recientemente por Intel. Esta tecnología supone algo parecido a tener dos procesadores lógicos para proceso en paralelo; aunque los programas deben ser compilados para aprovecharse de esto. Además, con ellos se introdujo la denominada arquitectura *Netburst*. Las últimas versiones incluyen cachés de nivel 2 de gran tamaño, por lo que están orientados, fundamentalmente, a tareas multimedia (edición de foto y vídeo digital), juegos y software de gestión (bases de datos, facturación, etc.). Las frecuencias de reloj con las que aparecen en el mercado los diferentes modelos de este procesador oscilan entre los 2,40 y los 3,80 GHz. Intel sacó al mercado una versión especial del Pentium IV, dotada con 2 MB adicionales de caché de nivel 3 y denominada *Extreme Edition*, con el objetivo de aumentar el rendimiento de las aplicaciones multimedia con alta demanda de gráficos (las "malas lenguas" dicen que se trata de un procesador específicamente pensado para juegos). La verdad es que Intel fue bastante criticado por ello y, ciertamente, no vale la pena pagar la diferencia de precio a no ser que se sea un jugador profesional[9].

- *Xeon*: éstos son procesadores de la gama Pentium, pero mejorados. Por ejemplo, admiten funcionar conjuntamente con otro u otros procesadores conectados a la misma placa y presentan características ampliadas; normalmente más memorias de caché y de mayor tamaño. Esta serie de procesadores se creó para su uso en servidores y estaciones de trabajo de alto rendimiento. En general, tienen tres niveles de memoria de caché y éstas son de gran tamaño. Se les considera procesadores de gama alta, ya que incluyen lo último en tecnología y, como hemos dicho, tienen una rama multiprocesador que admite la conexión de varios procesadores en la misma placa. Se ofrecen modelos de Xeon (algunos ya obsoletos) que abarcan frecuencias de reloj entre 1,5 y 3,73 GHz. Últimamente, han salido al mercado versiones de 64 bits (pero compatibles con la arquitectura de 32 bits x86).

- *Core Duo*: en realidad, éste es el nombre de la tecnología de doble núcleo desarrollada por Intel, aunque se ha convertido también en la denominación que se le da a sus procesadores de última generación que la emplean. Con esta tecnología, el encapsulado del procesador incluye dos núcleos *físicos* (reales) de ejecución. Lo que se consigue, básicamente, es una notable mejora en el rendimiento y prestaciones en los sistemas multitarea, ya que puede ejecutar varias aplicaciones exigentes simultáneamente, como juegos con gráficos potentes o programas que requieran muchos cálculos. Puesto que los sistemas empleados hoy en día son multitarea, la mejora está servida; y, de hecho, se notará aún más en la medida en que estemos trabajando simultáneamente con varias aplicaciones (y, eso sí, siempre que la memoria RAM tenga suficiente capacidad para albergarlas y no tengamos que echar mano de la memoria de intercambio o memoria *swap*[10]). En el mercado se encuentran versiones con frecuencias de reloj que oscilan entre los 1,06 y los 2,33 GHz. Sin embargo, debe hacerse constar que estos procesadores

[9] Existen competiciones organizadas de juegos con premios en metálico, como los *World Cyber Games* (`http://en.wikipedia.org/wiki/World_Cyber_Games`).

[10] Ya explicaremos qué es esta clase de memoria en el apartado 4.3.10.

son de *32 bits*. La mejora casi inmediata de esta tecnología por parte de Intel, trajo consigo las versiones *de 64 bits Core 2 Duo* (también para portátiles) con más prestaciones (hasta 3 GHz) y *Core 2 Duo Extreme* (hasta 3,16 GHz y más caché).

- *Core 2 Duo/Quad*: la tecnología Core 2 es la sucesora de la tecnología *Dual Core* original que permitía incluir dos núcleos separados de ejecución en el mismo encapsulado del procesador. Supone una evolución ya que emplea núcleos de ejecución de *64 bits* y, por tanto, mejora sobre la tecnología original, sobre todo en cuanto a consumo energético y mayor velocidad de proceso (hasta un $40\,\%$ más). La última gama de procesadores de Intel, *Quad*, presenta cuatro núcleos de ejecución en el encapsulado, incrementando notablemente el rendimiento de la multitarea y el paralelismo en los ordenadores que actualmente los incluyen. Se ofrecen velocidades de reloj entre 1,06 y 3,16 GHz.

- *Mobile/Centrino*: éste es el nombre que originalmente recibía la gama de procesadores Pentium para portátiles. Hoy en día, el "sello" Centrino se aplica a las diferentes versiones de procesadores, de cualquier gama, específicas para ordenadores portátiles. En realidad, Centrino hace referencia a aquellos ordenadores portátiles que incluyen una "combinación" de procesador Mobile de Intel (o Core Duo para portátil), junto con un chipset específico de Intel y una tarjeta ethernet Wi-fi para conexión a una red inalámbrica. Este tipo de procesadores destacan por su bajo consumo, por lo que el equipo desprende mucho menos calor y tiene una mayor autonomía, ya que la duración efectiva de la batería es mayor. Además, los procesadores de esta gama emplean una tecnología de encapsulado específica. Otra de sus características destacadas es que incluyen una memoria caché de nivel 2 de gran tamaño. Las frecuencias de reloj típicas de estos procesadores oscilan entre 1,30 y 2,66 GHz. Aunque tienen menos velocidad, su rendimiento es similar al de los Pentium 4 o Core Duo para ordenadores de sobremesa.

- *Itanium/Itanium 2*: esta gama de procesadores no puede considerarse realmente como propia de la arquitectura x86, ya que se trata de una arquitectura nueva pensada para trabajar con un ancho de 64 bits y con un nuevo juego de instrucciones exclusivamente de 64 bits, por tanto, rompe la compatibilidad con la típica arquitectura PC. La arquitectura del Itanium se diferencia drásticamente no sólo de la arquitectura x86, sino también de su extensión a 64 bits usada en otros procesadores de Intel. Se basa en la utilización de paralelismo explícito *a nivel de instrucción*, siendo el compilador el que toma las decisiones acerca de qué instrucciones han de ejecutarse en paralelo. Esta aproximación permite que el procesador sea capaz de ejecutar hasta seis instrucciones por ciclo de reloj. En contraste con otras arquitecturas superescalares, los procesadores de la gama Itanium no usan circuitería refinada para resolver las dependencias entre instrucciones durante la ejecución en paralelo de las mismas, ya que es el compilador el que debe dar cuenta de ellas al construir el ejecutable. Este tipo de procesadores está pensado para su empleo en servidores de altas prestaciones. En la actualidad, podemos encontrar procesadores con una frecuencia de reloj que oscila

entre 1,3 y 1,66 GHz y también con diferentes tamaños de caché (aunque son, en general, muy grandes).

La empresa AMD es el otro fabricante importante de procesadores para PC. Es el principal rival de Intel y entre ambos copan el mercado. Destacamos las siguientes gamas de procesadores:

- *Athlon XP*: aunque actualmente ya no se fabrican, estos procesadores supusieron la tercera generación de la gama de procesadores K7, poco antes de quedar obsoleta ante los nuevos avances tecnológicos en el campo de los microprocesadores. Posteriormente, hubo una cuarta y quinta generación de K7, pero ninguno gozó de la popularidad de este procesador en equipos de sobremesa. Hoy en día, todavía podemos encontrar estos procesadores en equipos de hace unos pocos años (de 2002 en adelante). Estos procesadores competían con los Pentium IV y ofrecían una gran capacidad multimedia, ya que incluían tanto el juego de instrucciones multimedia (mejorado) *3DNow! Professional* propio de AMD, como el SSE (*Streaming SIMD Extensions*) de Intel. Con respecto a sus predecesores ofrecía un $10\,\%$ más de rendimiento y un $20\,\%$ menos de consumo energético[11], el auténtico punto débil de la gama de procesadores K7 hasta ese momento. Este procesador poseía una caché de primer nivel (L1) de 64 KB y otra de segundo nivel (L2) de 256 KB. Salió al mercado con frecuencias de reloj que oscilaban entre $1{,}3$ y $1{,}7$ GHz (en el rango de modelos que iba del 1500+ al 2100+). Parece poco en comparación con las velocidades de su rival, el Pentium 4, pero hay que tener en cuenta que los procesadores de AMD realizaban en torno a un $150\,\%$ más de trabajo en cada ciclo de reloj. Posteriormente, con la cuarta y quinta generación de K7, se llegó a alcanzar los 2 y los $2{,}33$ GHz respectivamente. Se sacó al mercado una gama específica para portátiles, *Mobile Athlon XP*, con capacidad de ajustar la frecuencia de reloj de forma dinámica en función de la carga del procesador, con lo cual se reducía el consumo y se evitaba el sobrecalentamiento del portátil.
- *Athlon 64*: este procesador es el sucesor de la gama Athlon XP para el mercado doméstico. Es un procesador de 64 bits que no sólo soporta aplicaciones de 64 bits (ya que incluye parte del juego de instrucciones de Opteron) sino que también mantiene la compatibilidad con las de 32 bits. Es un procesador de gran potencia, ideal para juegos y aplicaciones con uso intensivo de gráficos en 3D. Su potencial ha sido desaprovechado en parte por el *software* de uso común, aunque esta situación ha cambiado hoy en día. Sus frecuencias de reloj típicas oscilan entre 2,2 y 2,6 GHz y presentan una memoria de caché de nivel 2 entre 512 KB y 1 MB. AMD fabricó una versión de este procesador orientada a juegos y multimedia de mayor potencia (con frecuencias de reloj entre 2,6 y 2,8 GHz) denominada *FX*. También existe una versión de doble núcleo (similar a la tecnología “Core Duo” de Intel) con mayor potencia y caché y velocidades entre 2 y 3,2 GHz.

[11]Con lo cual el procesador tardaba más en calentarse y lo hacía en menor grado.

- *Opteron*: es un procesador con arquitectura de 64 bits, pero compatible con aplicaciones de 32 bits sin merma de velocidad. Los procesadores de esta gama están orientados a servidores y a computación de altas prestaciones. Se comparan con Itanium 2 y la gama alta de Pentium Xeon (64 bits). Presentan frecuencias de reloj entre 1,6 y 3 GHz y tienen también una versión de doble núcleo.
- *Sempron*: es la gama de procesadores de AMD de bajo coste (y menos prestaciones) que compiten con los Celeron D de Intel. Son procesadores de gama media-baja y tienen menos prestaciones que los Athlon 64, ya que presentan memorias de caché de un tamaño entre dos y cuatro veces menor y originalmente no soportaban 64 bits, aunque los modelos actuales (a partir de la segunda mitad de 2005) sí que soportan aplicaciones de 64 bits (la arquitectura mixta conocida como x86-64) y han heredado muchas de las características de los Athlon XP (por ejemplo, el juego de instrucciones multimedia 3DNow!). Las frecuencias de reloj habituales para este tipo de procesadores oscilan entre 1,6 y 2,3 GHz.
- *Mobile Athlon 64/Sempron* y *Turion*: constituyen las gamas de procesadores de AMD específicos para portátiles. Son la competencia de los modelos Pentium Centrino, Mobile y Core 2 Duo del fabricante Intel. Este tipo de procesadores son de bajo consumo y mantienen altas (Athlon 64) o medias (Sempron) prestaciones. Los Athlon 64 presentan velocidades entre 1,6 y 2,6 GHz, soportan 64 bits y tienen 2 ó 4 veces más caché (nivel 2) que los Sempron. En los *Turion* encontramos dos versiones: *Turion 64* y *Turion 64 X2*. Ambos son procesadores de 64 bits y menor consumo, con velocidades (frecuencias de reloj) entre 1,6 y 2 GHz; aunque los segundos tienen mayores prestaciones ya que son de doble núcleo.

A la vista de lo que hemos expuesto hasta el momento, el lector podría preguntarse hasta qué punto cabe esperar que avance la tecnología en el campo de los procesadores. Ésta es una cuestión sin duda interesante y, de hecho, ya se ha venido planteando desde hace muchos años. Concretamente en 1965, Gordon Moore, uno de los cofundadores de la empresa Intel, observó en un artículo que el número de transistores que se pueden situar en un circuito integrado crecía exponencialmente, doblándose aproximadamente cada dos años.



Figura 4.21. Ventilador sobre la superficie de un procesador

A esta observación, casi escrupulosamente cumplida empíricamente a lo largo de más de 35 años, se le conoce popularmente con el nombre de *ley de Moore* y su enunciado viene a decir que la densidad de los circuitos integrados de los ordenadores se duplica cada 18 meses, aproximadamente. Se estima que la ley de Moore se mantendrá, al menos, durante la próxima década y, tal vez, mucho más.

Sin embargo, los principales fabricantes de procesadores empiezan a encontrarse con algunos problemas colaterales en la actualidad. A principios de siglo, Intel apostaba por incrementar la *frecuencia de reloj*, mientras que AMD apostaba por incrementar el *rendimiento* con incrementos bajos de frecuencia. Hacia el 2005, nos encontramos con que Intel tenía serios problemas para aumentar la velocidad más allá de los 3,8 GHz (problemas relacionados con la estabilidad y el calentamiento), mientras que los procesadores de AMD se calentaban en exceso: ¡comienza a haber problemas! La solución aplicada ha consistido en aumentar el *paralelismo* (llevando al límite la técnica de *hyperthreading*) y mejorar la tecnología de fabricación. Las tecnologías más en boga hoy en día son las de *doble núcleo* (*dual core*) y la denominada *x86-64*, que no es más que la extensión de la arquitectura x86 de 32 bits para emplear una *arquitectura interna de 64 bits* (pero compatible con aplicaciones de 32 bits).

4.3.7. Ventilación de la caja

Los procesadores, tanto de Intel como de AMD[12], suelen disipar mucho calor durante su funcionamiento. Recientemente, se han realizado muchos avances en este sentido. Los procesadores que se comercializan en la actualidad (en particular, los específicos para portátiles) han reducido su consumo energético en relación a los procesadores existentes en el mercado hace tres años. Cabe recordar que un menor consumo energético suele traer consigo un menor calentamiento de los materiales. Por otro lado, emplean tecnologías de disipación de calor bastante efectivas. Aun así, todos los procesadores requieren una buena ventilación para su correcto funcionamiento.



Figura 4.22. Un curioso sistema de refrigeración por agua

Sobre la superficie del procesador se suele disponer un radiador y, sobre él, un ventilador o disipador térmico. Esto es así, no sólo con los procesadores convencionales de ordenador (CPU), sino que también ocurre con los procesadores específicos de las tarjetas gráficas (GPU, del inglés *Graphics Processing Unit*), por ejemplo.

Una de las cosas que los *hackers* de los ordenadores personales suelen hacer (y que estuvo "de moda" hace unos cuantos años) es incrementar la frecuencia del reloj interno (el del procesador) dentro de ciertos márgenes. Esto es lo que se conoce popularmente con el nombre de *overclocking*. De esta forma, es posible obtener un mayor rendimiento de nuestro ordenador sin necesidad de cambiar el procesador.

Esto es posible porque el reloj interno del procesador lo que hace, en realidad, es multiplicar la frecuencia del reloj externo (el de la placa base). Por defecto, muchos modelos de procesador permitían aumentar el factor de multiplicación en una o dos

[12] Y también los de otros fabricantes.

unidades (mediante un sencillo *hack*) sobre el valor estándar establecido de fábrica. Sin embargo, se corren riesgos: algunos procesadores comenzaban a fallar más de la cuenta y tenían comportamientos "extraños" y, por otro lado, se aumenta en gran medida el calor generado y el procesador puede quemarse. Incluso se conoce de la existencia de sistemas de refrigeración ¡por agua!, tal como muestra la Figura 4.22.

Además del ventilador o disipador del procesador, podemos encontrarnos con otras fuentes de ventilación y refrigeración en un ordenador personal, tal como es el caso ya mencionado de las GPU de algunas tarjetas gráficas con aceleración 3D, o el de la fuente de alimentación, e incluso los de algunos discos duros. Además, suelen disponerse estratégicamente colocados uno o dos ventiladores adicionales en la caja del ordenador (según las dimensiones de la caja y el número de elementos dispuestos en su interior) para lograr una mayor disipación del calor generado.

En cualquier caso, se debe tener en cuenta que los ventiladores son una *fuente de ruido* sonoro considerable. Precisamente por ello, se fabrican cajas que incluyen paneles especiales para absorber el ruido, fuentes de alimentación silenciosas y juntas de goma para aislar el ruido de los dispositivos internos, entre otros, tal como vimos en la sección 4.3.1.

4.3.8. El zócalo (socket) del microprocesador

El *socket* donde se sitúa (conecta) el microprocesador (denominado en castellano zócalo) consiste en una matriz de pequeños agujeros existente en la placa base donde encajan, sin dificultad, los pines o patillas del procesador. Dicha matriz, denominada *Pin Grid Array* (PGA), permite la conexión entre el microprocesador y dicha placa base. Ten en cuenta que, cuando hablamos de *pines* (patillas) de un microprocesador, nos estamos refiriendo a cada uno de los contactos terminales de un conector o componente electrónico, fabricado de un material conductor de la electricidad.

En los primeros ordenadores personales, el microprocesador venía directamente soldado a la placa base, pero la aparición de una amplia gama de microprocesadores llevó a la creación del *socket*. En general, cada familia de microprocesador requiere un tipo distinto de zócalo, ya que existen diferencias en el número de pines, su disposición geométrica y la interconexión requerida con los componentes de la placa base. Por tanto, no es posible conectar un microprocesador a una placa base con un zócalo no diseñado para él.

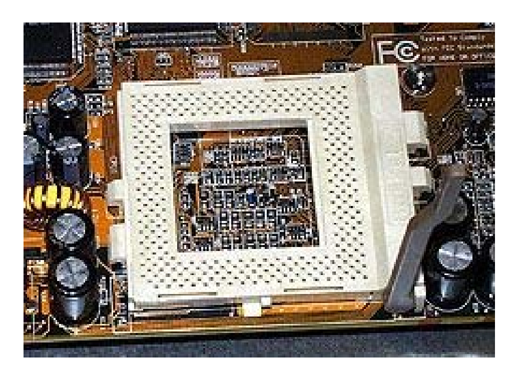


Figura 4.23. El *socket* 370

En la actualidad existen varios modelos de zócalo PGA, como por ejemplo el *socket 775*[13] (para procesadores Intel de la gama Pentium 4, Pentium D, Core 2 Duo y Core 2 Quad) , el *socket 479* (para procesadores Intel Pentium o Celeron Mobile) o el *socket AM2* (para procesadores AMD de las gamas Athlon 64, Athlon 64 X2, Athlon 64 FX y Opteron). La Figura 4.24 muestra

[13]También conocido como *socket T*.

imágenes de un *socket* PGA.

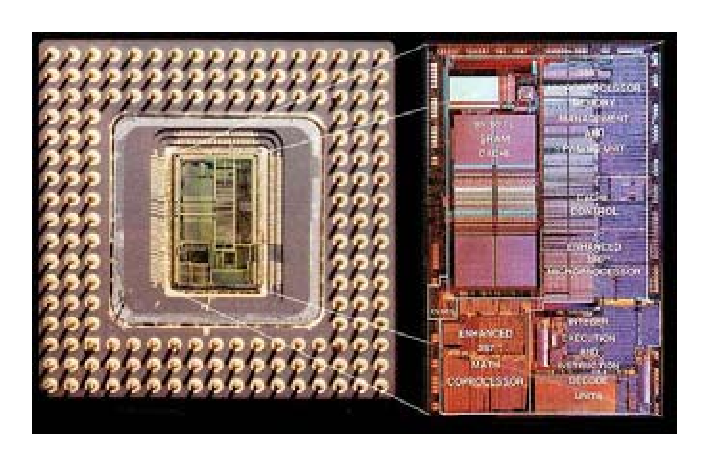




Figura 4.24. En la parte izquierda vemos un primer plano del *socket* para un 486 (Intel); a la derecha vemos dicho *socket* ubicado en la placa base (parte superior)

Como dato interesante, queremos comentar que, en algunas ocasiones, se ha podido encontrar conexiones en la placa con forma de *ranura* para el procesador. Esto ocurrió hace muchos años: con la introducción del Pentium II, Intel presentó un nuevo tipo de empaquetado para los microprocesadores; el denominado *SECC* (del inglés *Single Edge Contact Cartridge*).







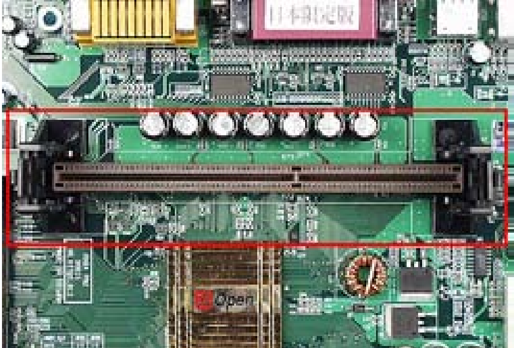


Figura 4.25. Varias ilustraciones de *sockets* tipo SECC. Arriba vemos sendas imágenes de procesadores Pentium II de Intel. Abajo a la izquierda: un modelo de Pentium III empaquetado en un SECC. Abajo a la derecha: una ranura tipo SECC para conectar un procesador en la placa base

Otros procesadores posteriores emplearon un empaquetado similar y también se

conectaban en ranuras de la placa base. El SECC agrupaba tanto el microprocesador como la memoria caché. El zócalo pasó entonces a ser una ranura: originalmente era el denominado *Slot 1*.

Cuando Intel introdujo en el mercado el procesador Pentium Xeon, el cual se trataba de una gama de Pentium con mayores prestaciones (entre ellas, más memoria caché) se necesitó un cartucho más grande, así que Intel diseñó un nuevo zócalo, el denominado *Slot 2*.

También la empresa rival, AMD, introdujo *sockets* con forma de ranura para algunos de sus modelos de procesador. Podemos mencionar, por ejemplo, el *Slot A*, similar al *Slot 1*, pero incompatible; es decir, que los procesadores que se conectaban a un *Slot 1* no podían conectarse en un *Slot A* y viceversa. Las imágenes de la Figura 4.25 muestran algunos ejemplos de *sockets* con forma de ranura.

Ya en 1999, Intel reintrodujo el socket PGA al presentar un nuevo modelo de procesador Pentium de prestaciones reducidas: el *Celeron*. Ese nuevo *socket* PGA se conoció por el nombre de *Socket 370*, del cual podemos ver una imagen en la Figura 4.23.

Posteriormente, Intel presentó otras versiones del zócalo PGA, como, por ejemplo, el *FC-PGA* (Flip Chip-PGA) y el *FC-PGA2*. En los albores del nuevo siglo (año 2000) y coincidiendo con la salida al mercado del Pentium 4, Intel volvió a presentar otro tipo de zócalo PGA: el *Socket 423*. Y así han sido surgiendo nuevos tipos hasta llegar hasta nuestros días (pueden consultarse en `http://en.wikipedia.org/wiki/CPU_socket` y `http://es.wikipedia.org/wiki/Lista_de_sockets`).

Si decides montarte un ordenador (PC) por piezas, debes asegurarte de que el procesador escogido puede instalarse en el zócalo disponible en la placa base sobre la que vas a montar los diferentes componentes.

4.3.9. Los conectores

En la placa base encontramos una serie de "pistas" de comunicación entre dispositivos: son los buses. Algunos de estos buses están pensados para la conexión de ciertos dispositivos desde el exterior (discos duros externos, cámaras de fotos y de vídeo, teclados, ratones, impresoras o escáneres, por ejemplo). Es por ello que existen ciertos conectores en la placa que permiten establecer una conexión con los buses integrados en la misma.

A estos conectores se les denomina, habitualmente (y coloquialmente), *puertos* y, normalmente, los podemos encontrar sobresaliendo en la parte trasera de la caja, tal como indica la figura de la derecha. Hoy en día, y por comodidad, podemos encontrarlos habitualmente en la parte frontal, cerca de los botones de encendido y RESET (nos estamos refiriendo fundamentalmente a los puertos USB).



Figura 4.26. Vista trasera de la caja: en la parte superior izquierda distinguimos, de arriba a abajo, dos puertos PS2 para teclado (morado) y ratón (verde), dos puertos USB, dos puertos serie (en verde) y un puerto paralelo (magenta)

En la Figura 4.26 puedes apreciar que estos conectores presentan diferente aspecto

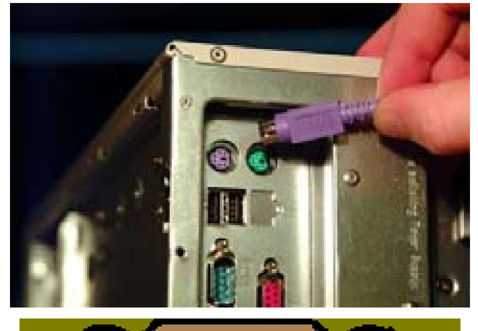
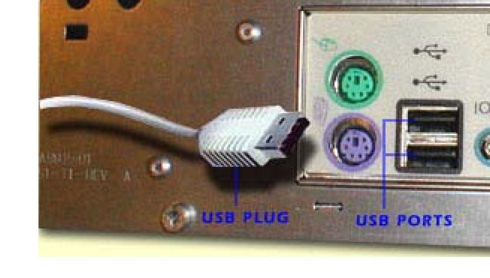







Figura 4.27. Diversos tipos de conectores en el PC. Arriba vemos imágenes de puertos PS2 (izquierda) y USB (derecha). Abajo a la izquierda: dibujo de un puerto serie. Abajo a la derecha: un puerto paralelo

según su tipo (serie, paralelo, USB, etc.). Debes tener presente que cada uno permite conectar cierto tipo de dispositivos. Por ejemplo, el puerto serie permitía[14] conectar ratones y módems telefónicos convencionales; el puerto paralelo, impresoras; los puertos USB (supongamos USB 2.0), discos duros y cámaras de fotos y vídeo; los puertos de tipo PS2 son específicos para teclado y ratón, etc.

La Figura 4.27 muestra varios tipos de conectores; aquéllos que con mayor asiduidad podemos encontrar en los ordenadores personales en la actualidad. Prácticamente todos los PC (de sobremesa y portátiles) incluyen conectores de esta clase. No obstante, queremos hacer notar que los puertos serie y paralelo (salvo para la conexión de ciertos dispositivos "especiales") están ya prácticamente obsoletos. También ocurre algo parecido con los puertos PS2, pues hoy en día es común conectar los teclados y ratones a puertos USB de baja velocidad.

Tal como hemos comentado, los conectores de tipo PS2 sirven para conectar el teclado y el ratón a nuestro ordenador. El puerto serie es un puerto de baja velocidad: en torno a 115 Kbps. y es un puerto a extinguir, aunque suele venir al menos uno en todos los ordenadores. Se usa(ba), principalmente, para conectar módems telefónicos convencionales[15] externos y también ratones. Últimamente, dispositivos como el módem se conectan a un puerto USB o bien son internos, y los ratones se conectan a los puertos PS2 o USB.

El puerto paralelo se usa(ba), principalmente, para conectar impresoras. En la actualidad, la mayoría de las impresoras se conectan al puerto USB de alta velocidad (USB 2.0). El paralelo es un puerto lento, aunque más rápido que el serie: funciona a 1 Mbps. Además, también se podían conectar ciertos dispositivos de almacenamiento externo como las unidades ZIP de puerto paralelo (100 MB), hoy en día obsoletas (hay versiones de mayor capacidad para su conexión a USB).

Hoy en día, los puertos USB son los que se utilizan con mayor frecuencia para la

[14]Hoy en día está casi obsoleto, desde el punto de vista del usuario doméstico.
[15]Para distinguirlos de módems más específicos (y distintos), como los módems ADSL.

conexión de dispositivos externos al PC. USB son las siglas del inglés *Universal Serial Bus* (bus universal en serie). Este tipo de puertos es relativamente reciente (la primera especificación data de mediados de los noventa, en el siglo pasado). En la actualidad, los ordenadores suelen llevar tres o cuatro puertos USB preinstalados (un par de baja velocidad, 1.0 ó 1.1, y otro par de alta velocidad, 2.0, normalmente).

4.3.10. La memoria del computador

En los ordenadores personales actuales podemos encontrar diversos tipos de memoria, según su funcionalidad (almacén temporal de código de programas y datos, intercambio de datos. . .) y características (mayor o menor velocidad y/o capacidad). Su unidad de almacenamiento es el *byte*, el espacio que se necesita para almacenar un carácter, como ya sabemos. Hoy en día se mide en órdenes de MegaBytes (MB) o GigaBytes (GB). Simplificando, podemos distinguir, principalmente, cuatro tipos de memoria:

- La memoria principal, comúnmente conocida como memoria RAM (del inglés *Random Access Memory*). Es la memoria de trabajo del computador y alberga el código (las instrucciones) de los programas que están en ejecución junto con los datos que éstos manipulan. Es una memoria de gran capacidad; actualmente se dispone en módulos con circuitos del orden de los centenares de MB. Lo normal, hoy en día, es encontrarse equipos con 1 ó 2 GB de memoria RAM. De hecho, algunos sistemas operativos actuales (Windows Vista) requieren un mínimo de 1 GB para funcionar, aunque 2 GB es la cantidad recomendada para su óptimo funcionamiento.

- La memoria de caché de nivel 1 (también conocida por su forma abreviada caché L1). Este tipo de memoria se encuentra en el propio encapsulado del procesador y sirve para optimizar su rendimiento, tal como vimos en la sección 4.3.6. Como principales características destacan su enorme velocidad de acceso (es ultra-rápida) y su (relativamente) poca capacidad (del orden de los centenares de KB) si la comparamos con la memoria RAM, por ejemplo.

- La memoria de caché de nivel 2 (caché L2, en su forma abreviada). También tratamos de ella en la sección 4.3.6. Sabemos que puede ser interna o externa al encapsulado del microprocesador, que es algo menos rápida que la caché L1 y también algo mayor (del orden de 1 ó 2 MB habitualmente).

- La memoria virtual o memoria de intercambio (*swap*, en inglés). Esta clase de memoria es, en realidad, un artificio del sistema operativo. Consiste en emplear un fragmento del espacio en disco duro para usarlo como “ampliación” de la memoria RAM del sistema. El sistema operativo la usa habitualmente como zona de intercambio con los datos almacenados en la memoria principal. Resumiendo, podemos decir que cuando un programa lleva un determinado tiempo “ocioso” (sin ejecutar instrucciones), parte de la memoria RAM que ocupan sus datos y código es transferida a este espacio en disco, dejando un “hueco” libre en la RAM que pueden ocupar nuevos programas que entren en ejecución. Es extremadamente lenta, ya que los tiempos de acceso a disco son varios órdenes de

magnitud superiores a los que se emplean para el acceso a los circuitos de memoria RAM.

Queremos hacer notar al lector que, obviamente, no estamos tratando de las memorias de almacenamiento (masivo) secundario, tales como discos duros, memorias *Flash* USB, memorias *CompactFlash*, etc. Este tipo de memorias sirven de almacén permanente de datos y nosotros estamos tratando exclusivamente, ahora, de las memorias de trabajo del ordenador (aquéllas que intervienen y son necesarias para su funcionamiento cotidiano).



Figura 4.28. Módulos de memoria RAM en soporte DIMM

La memoria RAM se utiliza como memoria principal de trabajo del ordenador. En ella se almacena tanto las instrucciones que necesita ejecutar el microprocesador (el código de los programas en ejecución) como los datos que introducimos y deseamos procesar, así como los resultados obtenidos durante este proceso (esto es, los datos que manipulan los programas). Por tanto, se puede leer y también se puede escribir en ella. Tiene la característica de ser volátil, esto es, que sólo funciona mientras esté *encendido* el ordenador, ya que requiere alimentación eléctrica para poder almacenar información (si el ordenador se apaga, la información que contiene esta memoria se pierde). En adelante, nos ocuparemos sólo de la memoria principal.

Hoy en día, algunos procesadores de gama alta y altas prestaciones (tal como *Itanium 2*) incluyen circuitos adicionales de memoria de caché de nivel 3, que actúa entre la cache de nivel 2 y la memoria principal del ordenador. Esta memoria caché se puede implementar en un chip ajeno a la CPU y, normalmente, su tamaño oscila entre 2 y 256 MB. Los beneficios que puede aportar este tipo de cachés dependen de los patrones de acceso a memoria de las aplicaciones empleadas.

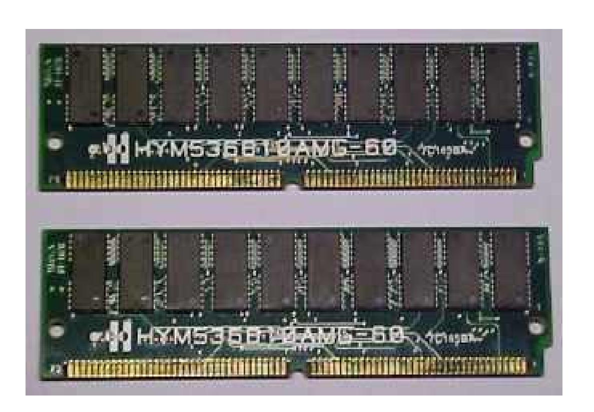


Figura 4.29. Módulos de memoria RAM en soporte SIMM (obsoletos)

También, los procesadores de gama alta y los pensados para ser utilizados en servidores incluyen circuitos de caché de nivel 3 en el propio encapsulado del procesador, incrementando la velocidad y reduciendo sustancialmente el coste. Por ejemplo, los procesadores Intel Xeon de última generación incluyen una caché de nivel 3 de 16 MB en el encapsulado del procesador, que es compartida entre los dos núcleos de ejecución.

A medida que la complejidad y la funcionalidad de los programas de aplicación ha ido aumentando, la utilización de la memoria RAM ha cobrado cada vez

mayor importancia. En la actualidad, podemos afirmar que la memoria RAM es un componente muy importante y, en la práctica, determinante para la velocidad del ordenador (por ejemplo, un sistema operativo como *Windows Vista* requiere de 2 GB de RAM para un funcionamiento óptimo). Podríamos decir que cuanta más memoria RAM mejor, ya que así, entre otras cosas, el sistema operativo tiene que recurrir menos a usar la memoria virtual (que ya sabemos que es muy lenta, en comparación). Algunos "visionarios" han tenido que comerse sus palabras con el paso del tiempo: Bill Gates afirmó al respecto de las limitaciones de MS-DOS (el primer S.O. para PC comercializado por Microsoft) con el direccionamiento de la memoria RAM que "640 KB deberían ser suficientes para cualquiera". Hoy en día, lo normal es instalar entre 1 y 2 GB de memoria RAM.

Los circuitos de memoria se disponen en módulos que se acoplan en algún tipo de soporte. Dicho soporte se encaja en una ranura de la placa base. Normalmente, las placas base incluyen varias ranuras en las que se disponen los soportes de memoria. Podemos distinguir dos grandes tipos de soportes o módulos en los que se distribuyen los circuitos de memoria:

- *SIMM*: fueron los primeros tipos de módulos de memoria que se podían insertar en zócalos (inicialmente la expansión de memoria se hacía insertando directamente los chips en los zócalos). Están obsoletos en la actualidad. Soportaban entre 256 KB y 32 MB. Los había de 30 y de 72 contactos (según el tipo de ranura a la que se conectaban en la placa base). Permitían gestionar 32 bits en cada acceso; es por ello por lo que se solían disponer dos zócalos "emparejados" para poder direccionar 64 bits por acceso (los zócalos se agrupaban en bancos que debían estar llenos de SIMM del mismo tipo; lo cual limitaba las posibilidades de ampliación de memoria).
- *DIMM*: es el tipo de soporte que se usa hoy día. Es más reciente y con mayor capacidad: en la actualidad se venden módulos que soportan entre 256 MB y 4 GB (y, al contrario que los SIMM, pueden colocarse en cualquier combinación). Este tipo de módulos permiten gestionar 64 bits en cada acceso. Más recientemente se ha desarrollado otro tipo de soporte basado en éste; es el denominado SO-DIMM. Éste consiste en una versión más compacta del DIMM convencional, con menos contactos y con aproximadamente la mitad de tamaño. Está orientado fundamentalmente a portátiles. Dispositivos como las agendas electrónicas (PDA, del inglés *Personal Digital Assistant*) usan, normalmente, memorias *CompactFlash*, MMC (*MultiMedia Card*) o SD (*Secure Digital)*).

Según la tecnología empleada para el diseño de los circuitos de memoria que se disponen en un soporte determinado, podemos encontrar diferentes tipos de memoria RAM. Se diferencian, fundamentalmente, por la velocidad de acceso, la cual se mide por el tiempo de respuesta, normalmente en nanosegundos (cuanto menos mejor). Cada placa base acepta memoria de uno o dos tipos únicamente.

La memoria que se emplea actualmente en los ordenadores personales es de tipo DRAM (*Dynamic Random Access Memory*). Ésta es una memoria RAM electrónica construida mediante condensadores. Los condensadores son capaces de almacenar un bit de información almacenando una carga, por lo que necesitan refrescarse cada cierto tiempo: el refresco de una memoria RAM consiste en recargar los condensadores que tienen almacenado un uno para evitar que la información se pierda (de ahí lo de dinámica). La memoria DRAM es más lenta que la memoria RAM estática (que suele emplearse en las memorias de caché tanto de la placa como de los discos), pero, por el contrario, es mucho más barata de fabricar y, por ello, es el tipo de memoria RAM comúnmente utilizada como memoria principal. Entre los tipos más recientes de memorias DRAM, podemos destacar los siguientes:

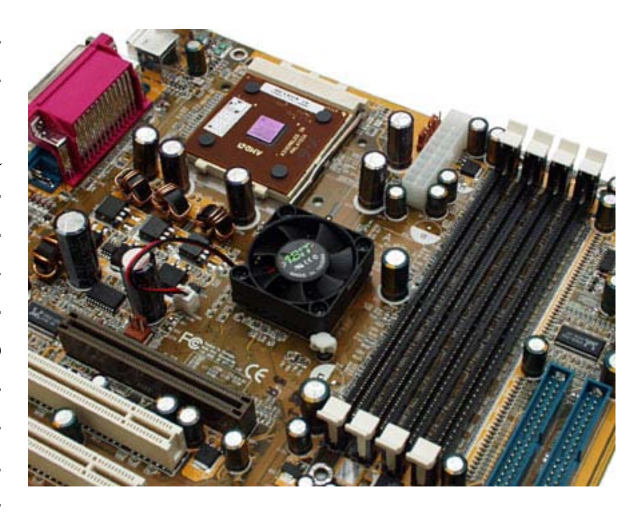


Figura 4.30. La memoria se monta sobre unas ranuras en la placa base

- SDRAM (por *Synchronous DRAM*). Es un tipo de memoria más rápido que el convencional DRAM, que era asíncrono. Gracias a la sincronía entre las diversas fases de acceso a memoria se consiguen evitar los ciclos de espera.
- ESDRAM (por *Enhanced SDRAM*). Este tipo de memoria aumentaba las prestaciones de la memoria DRAM al incluir una pequeña memoria estática auxiliar en el interior del chip SDRAM. El funcionamiento de esta memoria estática se basaba en un principio muy similar al de la memoria de caché utilizada en los procesadores actuales: las peticiones de acceso a lugares recientemente accedidos podían ser resueltas por esta rápida memoria.
- DDR SDRAM (por *Double Data Rate SDRAM*). Es una memoria de doble tasa de transferencia de datos, ya que trabaja al doble de velocidad que el bus del sistema, puesto que permite la transferencia de datos por dos canales distintos simultáneamente en un mismo ciclo de reloj. Son módulos compuestos por memorias síncronas (SDRAM), disponibles en encapsulado DIMM. Comenzó a usarse masivamente como tecnología de memoria para ordenadores personales en el año 2000.
- DDR2 SDRAM. Hace unos pocos años se desarrolló una mejora (menor) de la tecnología DDR SDRAM: se trataba de la memoria DDR2 SDRAM, más rápida que la original, ya que funciona a mayores velocidades de reloj e introduce un mayor paralelismo, pero con mayor latencia (tiempos de espera). Los módulos DDR2 son capaces de trabajar con 4 bits por ciclo, es decir 2 de ida y 2 de vuelta en un mismo ciclo (permitiendo, por tanto, que durante cada ciclo de reloj se realicen cuatro transferencias), mejorando sustancialmente el ancho de banda

potencial bajo la misma frecuencia de una DDR tradicional. El mismo hecho de que el buffer de la memoria DDR2 pueda almacenar 4 bits, para luego enviarlos, es el causante de la mayor latencia, debido a que se necesita mayor tiempo de “escucha” por parte del buffer y mayor tiempo de trabajo por parte de los módulos de memoria para recopilar esos 4 bits antes de poder enviar la información. Este tipo de memoria se ve mejor aprovechada por procesadores basados en la tecnología de múltiples núcleos en el encapsulado (como los Dual Core –doble núcleo– de reciente aparición). En principio, requieren menos voltaje para funcionar y, por tanto, tienen un menor consumo de energía.

- DDR3 SDRAM. Constituye la última tecnología en memorias SDRAM y, actualmente, se están empezando a comercializar. Entre sus características destaca un mayor ancho de banda y velocidad de transferencia, e incluso menos consumo de energía que las memorias DDR2, ya que requieren menos voltaje para funcionar.

También existen algunas otras versiones que resultan algo más “exóticas”, basadas igualmente en la tecnología DRAM. Por ejemplo, tenemos la memoria DRDRAM (*Direct Rambus DRAM*), más costosa que la DDR SDRAM. Esta memoria fue utilizada originalmente por Intel para su Pentium 4, aunque fue rápidamente sustituida por la DDR SDRAM debido a su menor coste y similares prestaciones en la práctica. O también tenemos la memoria SLDRAM (por *Synchronous Link DRAM*), muy parecida en cuanto a prestaciones a la memoria DRDRAM, desarrollada por un consorcio de 20 fabricantes de la industria informática. Esta memoria era más barata que la *Rambus* al no requerir el pago de regalías que, por cuestiones de licencia, había que abonar al fabricar chips de memoria basados en la tecnología *Rambus*. SLDRAM es un estándar abierto que no requiere una nueva arquitectura (al contrario que DRDRAM) y que envía todas las señales por la misma línea, ahorrándose el tiempo requerido en la sincronización de múltiples líneas. En la práctica opera al doble de velocidad que el reloj del bus, al igual que la tecnología DDR SDRAM. Finalmente, queremos mencionar a la memoria RLDRAM (por *Reduced Latency DRAM*), que es un tipo de memoria DDR SDRAM que combina un acceso rápido y aleatorio, con un elevado ancho de banda. Este tipo de memoria está diseñado, principalmente, para su empleo en dispositivos utilizados en redes y para su uso en tecnologías de caché.

La integridad de los datos de la memoria se puede comprobar mediante *bits de paridad* y si se utilizan técnicas de corrección de errores pueden incluso corregirse algunos errores. Algunas memorias son capaces de detectar errores empleando los mencionados bits de paridad, y algunas otras (que son más caras) pueden corregir ciertos errores mediante técnicas de corrección (son las llamadas *memorias ECC*). Sin embargo, por cuestiones de precio, las memorias se compran normalmente sin paridad ni ECC.

4.3.11. Los buses

Los buses interconectan diferentes elementos del ordenador y les permiten intercambiar información. La información que se envía puede catalogarse en alguno de estos tres tipos:

- Datos propiamente dichos (con lo que se opera).
- Direcciones de memoria (a las que se accede para leer o escribir datos).
- Información de control (códigos binarios que pueden interpretarse como órdenes en los dispositivos conectados al bus).

En un ordenador personal actual podemos encontrarnos con varios tipos de buses, que pueden clasificarse en dos grandes tipos. Por un lado, tenemos los denominados *buses de expansión*, que son aquellos a los que conectamos tarjetas que ofrecen diversas funcionalidades, como, por ejemplo, procesar el sonido, recibir y procesar la señal de televisión, conectarse a Internet, o ampliar la capacidad de conexión a nuestro ordenador de otros dispositivos (discos duros, etc.). En este grupo tenemos el bus IDE/EIDE/ATA, el SATA, el PCI (incluyendo la nueva tecnología PCI Express) y el AGP (exclusivo para la conexión de tarjetas de vídeo).

Por otro lado, están los conocidos como *buses de periféricos*, que son aquellos que permiten la conexión directa de periféricos a nuestro ordenador, con frecuencia de almacenamiento secundario (discos duros, memorias *Flash* USB, memorias *CompactFlash*...), y que pueden ser externos (normalmente) o incluso internos. Entre ellos, podemos destacar al *Universal Serial Bus*, más conocido por USB (sus siglas), el estándar *IEEE 1394*, también conocido como *FireWire* (muy empleado para la conexión de cámaras de vídeo digital, para poder almacenar vídeos grabados en el disco duro para su posterior proceso), y el bus SCSI (empleado para la conexión de discos duros con grandes velocidades de transferencia *full duplex* en servidores de alto rendimiento).

A la hora de determinar el rendimiento que puede ofrecer un determinado tipo de bus, los parámetros que interesan son la *anchura* o número de bits que pueden manipularse simultáneamente y la velocidad de transferencia, que puede ser medida en Mbps. (mega*bits* por segundo) o en MBps. (mega*bytes* por segundo). Observe el lector que, para poder comparar, se ha de multiplicar la velocidad en MBps. (para pasar a Mbps.) o, alternativamente, dividir la velocidad en Mbps. (para pasar a MBps.), por 8.

El bus IDE/EIDE/ATA



Figura 4.31. Conectores para bus IDE/EIDE/ATA en la placa base

El bus del sistema IDE (del inglés *Integrated Drive Electronics*), también conocido por su forma más genérica como ATA (del inglés *Advanced Technology Attachment*) es un bus que permite controlar y conectar dispositivos de almacenamiento masivo de datos, tales como unidades de disco duro, al ordenador. En general, permite la conexión de dispositivos IDE y ATAPI (del inglés *Advanced Technology Attachment Packet Interface*):

discos duros, unidades de lectura/grabación de CD-ROM o DVD y regrabadoras[16] de CD-ROM o DVD.

El bus IDE está diseñado para conectarse a la circuitería de estos dispositivos, teniendo en cuenta que el controlador del dispositivo se encuentra integrado en su propia electrónica. El bus IDE original evolucionó hacia el bus EIDE (*Enhanced IDE*) y, posteriormente, al Ultra ATA, pero, básicamente, funciona igual (aunque más rápido). Con la última versión de Ultra ATA se pueden conseguir velocidades de transferencia en torno a 133 MBps. En la actualidad, y debido a la aparición del bus SATA (del inglés *Serial ATA*), del que hablaremos en la sección 4.3.11, al bus IDE/EIDE/ATA también se le conoce con el nombre de PATA (del inglés *Parallel ATA*). Las versiones más recientes de sistemas ATA que se han desarrollado y comercializado en el mercado hasta el momento son:

- ATA-1: primera especificación moderna.
- ATA-2: soportaba transferencias rápidas en bloque y acceso directo a memoria multipalabra (DMA por *Direct Memory Access*).
- ATA-3: constituyó una mera revisión del ATA-2.
- ATA-4: fue un avance tecnológico destacado, también conocido como Ultra-DMA o ATA-33, ya que soportaba velocidades de transferencia de hasta 33 MB-ps. (algo importante en su época).
- ATA-5: conocido como Ultra ATA/66, es una revisión del anterior que permite velocidades de transferencia de hasta 66 MBps.
- ATA-6: denominado Ultra ATA/100, ya que da soporte para velocidades de 100 MBps.
- ATA-7: es el estándar actual, también conocido como Ultra ATA/133, pues permite velocidades de hasta 133 MBps.

Desde la aparición del bus PCI (*Peripheral Component Interconnection*, véase la sección 4.3.11), las controladoras IDE casi siempre están incluidas en la placa base, inicialmente como un chip, para pasar posteriormente a formar parte del chipset. En los ordenadores personales suele haber dos buses IDE independientes, que se presentan como dos conectores separados. Cada bus IDE acepta la conexión de dos dispositivos, aunque presentan una limitación: uno tiene que estar configurado como *esclavo* y el otro como *maestro* para que la controladora sepa a/de qué dispositivo enviar/recibir los datos. Por tanto, en un ordenador equipado con bus IDE podemos conectar hasta 4 dispositivos, a elegir de entre discos duros, lectores de CD-ROM, DVD, grabadores, etc. La configuración de estos dispositivos como maestro o esclavo se realiza mediante *jumpers*, que son como conductores de pequeña longitud que se emplean para conectar dos o más pines y completar así un circuito. Habitualmente, un disco duro, lector de CD-ROM o dispositivo ATAPI en general puede estar configurado de una de estas tres formas:

[16]Son las unidades que permiten trabajar con CD y DVD regrabables.

- Como maestro (*master*). Si es el único dispositivo conectado a la controladora del bus, debe tener esta configuración, aunque a veces también funciona si está como esclavo. Si hay otro dispositivo, el otro debe estar obligatoriamente como esclavo.
- Como esclavo (*slave*). Ello implica que, necesariamente, debe haber otro dispositivo conectado al bus que actúa como maestro.
- Como selección por cable (*cable select*). El dispositivo será maestro o esclavo en función de su posición en el cable. Esto implica que, obligatoriamente, si hay otro dispositivo conectado al bus, también debe estar configurado como selección por cable. Si el dispositivo es el único que está conectado al bus, debe estar conectado con el cable en la posición de maestro. Normalmente, para distinguir el conector en el que se conectará el primer bus IDE se utilizan colores distintos.

El diseño IDE, basado en la conexión de dos dispositivos diferentes a un mismo bus, tiene el inconveniente de que mientras se accede a un dispositivo el otro dispositivo del mismo conector IDE no se puede usar. Este inconveniente, como veremos más adelante, está resuelto en SATA y en SCSI, que son buses que permiten usar dos dispositivos por canal.

El bus ISA

ISA (del inglés *Industry Standard Architecture*) fue el bus original de los primeros ordenadores personales, en concreto de la segunda generación sacada al mercado por IBM (serie AT). De hecho, en un comienzo se llamó bus AT, pero ante la aparición de otros buses, se decidió llamarlo ISA. A él se conectaban las denominadas *tarjetas de expansión*; dispositivos en forma rectangular con circuitos integrados, que permitían ampliar las capacidades del ordenador (para procesar sonido, gráficos, etc.) y que se conectaban insertando sus conectores en las ranuras de conexión al bus.

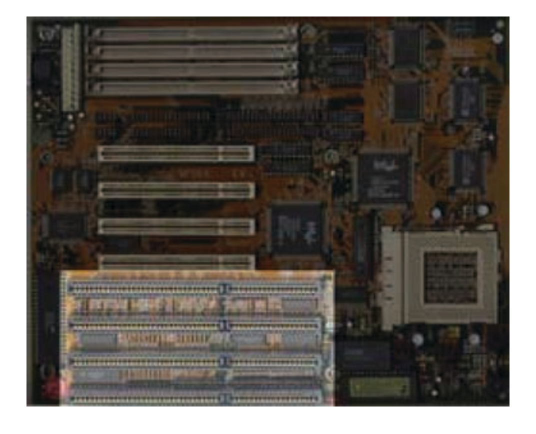


Figura 4.32. Los conectores para el bus ISA son fáciles de identificar en la placa base: son más grandes

Aunque ya está tecnológicamente obsoleto, es un bus que todavía podía encontrarse en las placas base que se comercializaban hasta hace unos pocos años por cuestiones de compatibilidad con muchas tarjetas (dispositivos) antiguas que exigían poco rendimiento (realmente, para ellos no suponía una pérdida de rendimiento): por ejemplo, éste era el caso de algunas tarjetas de sonido. Hoy en día, ya no se encuentra en las placas base pues las tarjetas de expansión se fabrican para su conexión al bus PCI, más reciente, potente, rápido y funcional que el ISA.

Este bus funcionaba a una velocidad muy lenta en comparación con otros componentes que se fueron desarrollando a lo largo del tiempo (a medida que avanzaba la tecnología): 8 MHz. Puesto que permitía transmitir 16 bits (éste era su ancho de banda)

simultáneamente (8 en un sentido y 8 en el otro) era capaz de transmitir aproximadamente 8 MBps.

Como curiosidad, queremos comentar que hubo intentos de mejorarlo, tales como el EISA (*Extended ISA*) que permitía velocidades de transferencia en torno a los 32 MBps. y el MCA (*Micro Channel Architecture*) que oscilaba entre los 40 y los 66 MBps., pero no tuvieron demasiado éxito y se vieron rápidamente postergados por la aparición del bus PCI.

El bus PCI

PCI (del inglés *Peripheral Component Interconnect*) es un bus de ordenador estándar que permite conectar dispositivos periféricos directamente a su placa base. Estos dispositivos pueden ser circuitos integrados ajustados en ésta o, lo más común, tarjetas de expansión que se ajustan en los conectores que ofrece el bus. Es el bus habitual en las placas base de los ordenadores personales, donde desplazó al ISA como bus estándar al ser mucho más rápido y funcional, aunque también se emplea en otro tipo de arquitecturas.

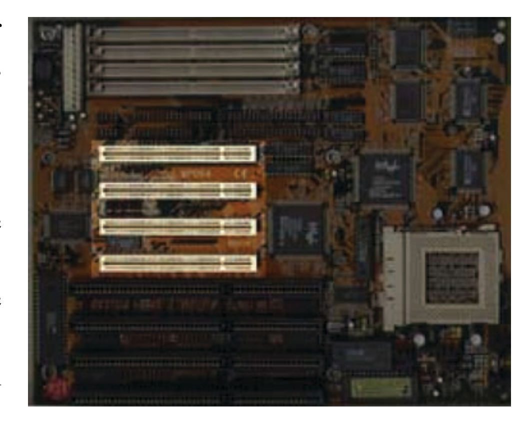


Figura 4.33. Conectores para el bus PCI en la placa base de un PC

A diferencia del bus ISA, el bus PCI permite la auto-configuración, esto es, la configuración dinámica de los dispositivos (tarjetas de expansión) periféricos que se le conectan: esto es lo que popularmente se conoce como *Plug & Play*. En el momento de arrancar el ordenador, justo antes del arranque del sistema operativo, las tarjetas conectadas al bus PCI y la BIOS interactúan y negocian los recursos solicitados por las tarjetas. Esto permite la asignación de IRQ (peticiones de interrupción) y direcciones para entrada y salida de datos, necesarios para la interacción entre sistema operativo (S.O.) y tarjeta, como veremos en la sección 4.4.2. Posteriormente, la BIOS pasa esa información al S.O. al arrancar éste.

Este proceso dinámico de configuración es diferente (y más funcional) al que empleaba el bus ISA original, donde las peticiones de interrupción (IRQ) tenían que ser configuradas manualmente usando *jumpers* externos. Precisamente para acabar con esta dificultad, las últimas revisiones del bus ISA y el entonces "nuevo" bus MCA de IBM ya incorporaron tecnologías que automatizaban todo el proceso de configuración de las tarjetas. Pero ya era tarde por entonces: el bus PCI demostró una mayor eficacia en la tecnología *Plug & Play* y ofrecía mejores prestaciones. Aparte de esto, el bus PCI proporcionaba una descripción detallada de todos los dispositivos PCI conectados a través de su espacio de configuración, característica ausente en los otros buses.

En cuanto a velocidad, el bus PCI ha sufrido diversas revisiones, y podemos decir que, según las especificaciones, nos encontramos con dos versiones del bus: la que tiene un ancho de banda de 32 bits y que ofrece una velocidad de transferencia de 133 MBps., y la que tiene un ancho de banda de 64 bits ofreciendo el doble de velocidad de transferencia: 266 MBps. El bus PCI trabaja habitualmente a una frecuencia de reloj

de 33,33 MHz y permite transferencias síncronas. Versiones posteriores, permitieron que el bus funcionase a 66 MHz. Las versiones PCI-X (PCI eXtendido) trabajan con frecuencias de 133, 266 y 533 MHz, permitiendo velocidades de transferencia a partir de 1014 MBps. y hasta rondar los 2 GBps.

En la actualidad, con la introducción del estándar PCI Express, a finales de 2004, los fabricantes de placas base han ido progresivamente eliminando ranuras de expansión para el bus PCI, sustituyéndolas por ranuras del nuevo estándar. En la actualidad, lo más común es que ambos tipos de buses estén presentes en las placas base de los ordenadores personales, si bien es previsible que el PCI vaya desapareciendo progresivamente en los próximos años.

El bus AGP

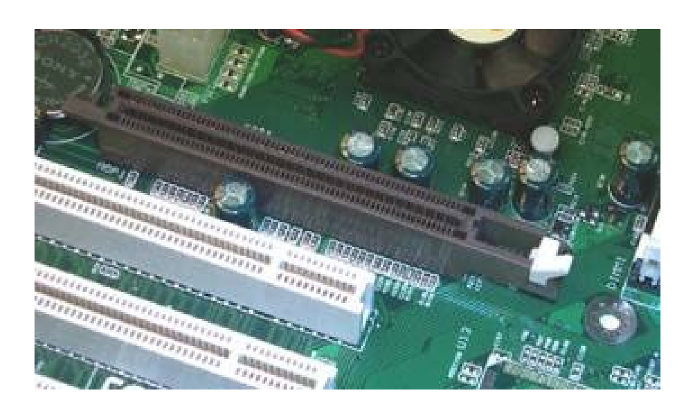


Figura 4.34. Conector para el puerto AGP en la placa base de un PC

El "bus" AGP (del inglés *Accelerated Graphics Port*, también conocido como *Advanced Graphics Port*) es un "bus" *dedicado exclusivamente a tarjetas de vídeo* (o tarjetas gráficas). Básicamente, se creó para reducir la carga que hasta hace unos años soportaba el bus PCI para trabajar con gráficos y permite, incluso, alcanzar mayores velocidades de transferencia y acceso directo a memoria.

El lector atento habrá notado, sin duda, la utilización del entrecomillado al referirnos al "bus" AGP. Esto es así porque, en puridad, no cabría calificar al AGP como un bus, sino específicamente como un *puerto*, ya que sólo permite la conexión de un único dispositivo, mientras que a un bus se pueden conectar varios. Hecha esta matización, nos referiremos en adelante a AGP indistintamente como bus o como puerto.

AGP fue originalmente desarrollado por Intel en el año 1996 como solución a los cuellos de botella (saturación) que se producían en las tarjetas gráficas que usaban el bus PCI. El diseño original partía de las especificaciones desarrolladas para la versión 2.1 del bus PCI. El puerto AGP es de 32 bits, como el PCI 2.1, pero cuenta con notables diferencias, entre las que destacan la utilización de 8 canales adicionales para acceso a la memoria RAM (para DMA o acceso directo a memoria). El acceso directo a memoria RAM permite que la tarjeta gráfica pueda almacenar en ella texturas y otros datos para dibujo en tres dimensiones, con lo que se consigue una mayor rapidez de ejecución en el uso de aplicaciones 3D (tales como CAD o juegos).

La frecuencia de reloj a la que funcionaba la primera versión de este bus era de 66 MHz y permitía una tasa de transferencia de 266 MBps. funcionando a un voltaje de 3,3 V. Con el tiempo aparecieron revisiones posteriores: AGP 2X, con frecuencia de 133 MHz y tasa de transferencia de 532 MBps.; AGP 4X, con frecuencia de 266 MHz, tasa de transferencia de 1 GBps. y funcionando a un voltaje de 3,3 o 1,5 V (para

adaptarse a los diseños de las tarjetas gráficas); y AGP 8X, con frecuencia de 533 MHz, tasa de transferencia de 2 GBps. y funcionando a un voltaje entre $0{,}7$ V y $1{,}5$ V.

Estas elevadas tasas de transferencias se consiguen aprovechando los ciclos de reloj del bus mediante un multiplicador, pero sin modificarlos físicamente. Como hemos comentado, el puerto AGP se utiliza exclusivamente para conectar tarjetas gráficas, y debido a su arquitectura sólo puede haber una ranura. Dicha ranura mide unos 8 cm. y se encuentra a un lado de las ranuras PCI, tal como se observa en la Figura 4.34.

A partir de 2006, el uso del puerto AGP ha ido disminuyendo con la aparición del nuevo bus estándar PCI Express, que proporciona mayores prestaciones en cuanto a frecuencia y ancho de banda. Así, los principales fabricantes de tarjetas gráficas, como ATI y nVIDIA, han ido presentando cada vez menos productos para este puerto. Las tarjetas gráficas de última generación de estos fabricantes ya salen al mercado casi exclusivamente para su conexión a PCI Express.

El bus PCI Express

El bus PCI Express, conocido en su forma abreviada como PCIE, constituye, básicamente, una *mejora* sobre el bus PCI, ya que presenta mayores velocidades de transferencia, con la idea de reemplazar a medio plazo a casi todos los demás buses, AGP y PCI incluidos. Debido a que se basa en el bus PCI, las tarjetas actuales pueden ser reconvertidas a PCI Express cambiando solamente la capa física.

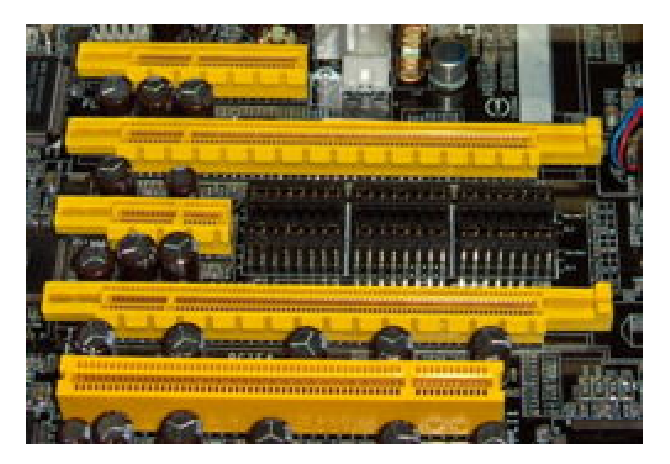


Figura 4.35. Conectores para el bus PCI Express en la placa base de un PC

PCIE está estructurado como una serie de enlaces punto a punto, funcionando en modo *full-duplex* y en serie. La versión 1.1 de PCIE (la más común en el año 2007) proporciona una velocidad de 250 MBps. en cada enlace (y, al ser full-duplex, independiente en cada sentido; es decir, que no se trata de una velocidad compartida entre ambos sentidos de la comunicación). La versión 2.0 de PCIE dobla esta tasa y parece ser que la versión 3.0 la dobla de nuevo (cuadruplicando la velocidad del 1.1).

Cada ranura de expansión para la conexión de tarjetas al bus lleva uno, dos, cuatro, ocho, dieciséis o treinta y dos enlaces de datos entre la placa base y las tarjetas conectadas. Para identificar el tipo de ranura, el número de enlaces se escribe con el prefijo x; así $x1$ indica un enlace simple y $x16$ se refiere a una tarjeta que requiere una conexión con dieciséis enlaces. Treinta y dos enlaces de 250 MBps. dan el máximo ancho de banda teórico del PCIE 1.1 en cada sentido de la comunicación: 8 GBps. $(250 \cdot 32)$.

El uso más habitual en las placas base actuales es $x16$, lo que proporciona un ancho de banda de 4 GBps. en cada sentido. En comparación con otros buses, un enlace simple es aproximadamente el doble de rápido que el PCI normal, una ranura

de cuatro enlaces ($x4$) tiene una velocidad de transferencia comparable a la versión más rápida del PCI-X 1.0, y una ranura de ocho enlaces ($x8$) tiene una tasa de transferencia comparable a la versión más rápida de AGP.

Frente a la pretensión de convertirlo en el bus único para PC, PCIE se encuentra con la desventaja de que no es todavía suficientemente rápido para ser usado como bus de memoria para la comunicación directa entre RAM y CPU. Actualmente se le considera como un estándar para la conexión de *tarjetas gráficas*. Los últimos modelos de nVIDIA y ATI son para PCI Express.

El bus SCSI

El bus SCSI (acrónimo del inglés *Small Computer System Interface*) es una interfaz estándar para la transferencia de datos entre distintos dispositivos del bus de la computadora. Para poder trabajar con un dispositivo SCSI en un ordenador personal (PC) es necesario que tanto el dispositivo como la placa base dispongan de un controlador SCSI, ya que, por defecto, SCSI no está presente en los PC. Lo normal es que el dispositivo venga con un controlador de este tipo. En lo que se refiere a la placa base, el controlador se puede incorporar mediante una tarjeta de expansión conectada, por ejemplo, al bus PCI. En la práctica, ocurre que hay diferentes tipos de conectores SCSI, lo cual puede resultar molesto en algunas situaciones. El bus SCSI necesita terminadores en los extremos, por cuestiones de funcionamiento del protocolo de comunicación en esta interfaz. La terminación del bus puede dar problemas, si bien existen muchos dispositivos SCSI con capacidad de *auto-terminación* de la cadena que los evitan.

La interfaz SCSI se utiliza habitualmente en los discos duros y los dispositivos de almacenamiento sobre cintas, pero también permite la interconexión de una amplia gama de dispositivos, tales como escáneres y regrabadoras de CD y DVD. De hecho, el estándar SCSI promueve la independencia de los dispositivos, lo que significa que, teóricamente, cualquier cosa puede ser hecha SCSI (incluso existen impresoras que utilizan la interfaz SCSI). El precio de la tarjeta SCSI y de los dispositivos compatibles con SCSI es elevado. Hoy se usa casi únicamente en ordenadores de altas prestaciones (servidores con uso intensivo de disco, por ejemplo).

En el pasado, era muy popular entre todas las clases de ordenadores, y también en los PC, ya que hace unos años sus buses eran extremadamente lentos (ni siquiera podían garantizar la velocidad de transferencia de datos necesaria para grabar un CD-ROM). El bus SCSI hacía posible conectar dispositivos que demandasen elevadas tasas de transferencia de datos (grabadoras de CD-ROM, por ejemplo).

Actualmente sigue siendo popular en puestos de alto rendimiento: servidores y periféricos de gama alta, como hemos comentado arriba. Sobre todo, continúan vendiéndose discos duros SCSI, ya que en servidores web donde se realizan muchas operaciones con el disco duro (y siendo éste el cuello de botella, por consiguiente) es importante obtener altas tasas de transferencia a disco y una elevada productividad. Los ordenadores de sobremesa y los portátiles utilizan habitualmente las interfaces más lentas (y baratas) IDE/EIDE/ATA o SATA para los discos duros y USB o FireWire (IEEE 1394) para la conexión de periféricos externos (discos, impresoras, cámaras...), a causa de la diferencia de coste. Recientemente, se ha desarrollado un sistema SCSI en serie, denominado *Serial Attached SCSI* (SAS), que, además, es compatible con

SATA: dado que utiliza el mismo conector se podrán conectar unidades SATA en una controladora SAS.

Podemos encontrarnos con varios tipos de SCSI: según la velocidad de transferencia tenemos desde los 5/10 MBps. de los primeros SCSI hasta los 640 MBps. ¡en *full duplex*! (ambos sentidos de transferencia) del Ultra-640 SCSI (año 2004); según el número máximo de dispositivos conectados "en cadena" (uno a continuación de otro) que soportan, tenemos entre 8 y 16 dispositivos habitualmente. Vamos a mencionar algunos de los más conocidos y populares:

- La primera especificación de SCSI fue el SCSI-1, también conocido como *Narrow SCSI*, a finales de los ochenta. Alcanzaba los 5 MBps. (muy alto para su época) y permitía la conexión de hasta 8 dispositivos en cadena.
- Debido a las mejoras tecnológicas en otros buses (como por ejemplo, la extensión EIDE del bus IDE), a mediados de los noventa aparece SCSI-2, popularmente conocido como *Fast SCSI*, que permitía alcanzar tasas de transferencia en torno a los 10 MBps.
- Poco después (finales de los noventa), surgen *Ultra SCSI* y *Fast-Wide SCSI*: ambos permiten alcanzar velocidades de 20 MBps., pero el segundo permite la conexión de hasta 16 dispositivos en cadena.
- Rápidamente llega una nueva mejora: *Ultra Wide SCSI* mantiene el tope de 16 dispositivos pero permite alcanzar los 40 MBps.
- Con la llegada del siglo XXI aparecen mejoras del interfaz: *Ultra2 Wide SCSI* (16 dispositivos, 80 MBps.) y *Ultra3 SCSI* que permite alcanzar los 160 MBps.
- Las últimas revisiones destacadas de la interfaz datan de los años 2003 y 2004: *Ultra-320 SCSI* (320 MBps.) y *Ultra-640 SCSI* (640 MBps.).
- En la actualidad, está creciendo la popularidad de la interfaz SAS (*Serial Attached SCSI*), que es una interfaz punto a punto y que no requiere terminador. SAS permite velocidades superiores a los 400 MBps. *en cada dirección*, a diferencia del SCSI paralelo tradicional. Permite la conexión de más de 16.000 dispositivos mediante el uso de expansores, lo cual le confiere, en la práctica, la posibilidad de conectar todos los dispositivos que queramos sin importar el número.

Los discos IDE/ATA están mucho más extendidos que los SCSI debido a su precio más bajo. Es cierto que el rendimiento de IDE/ATA es menor que el del SCSI, pero se están reduciendo las diferencias. En un ordenador casero (sobremesa) o portátil no haría falta usar SCSI: los dispositivos EIDE/ATA Ultra DMA (UDMA) alcanzan velocidades de transferencia en torno a los 133 MBps. Esto es así ya que el UDMA hace la función del Bus Mastering en SCSI con lo que se reduce la carga de la CPU y aumenta la velocidad. Y, si fuese necesario conectar dispositivos externos (por ejemplo, porque quisiéramos conectar una cámara de vídeo para capturar filmaciones), podemos incorporar una tarjeta de expansión *FireWire* (que puede verse como un tipo de SCSI) o bien usar un puerto de alta velocidad USB 2.0 (en ordenadores recientes). Los buses FireWire y USB 2.0 permiten alcanzar velocidades de transferencia aceptables para estos menesteres, aunque inferiores al IDE/ATA y, desde luego, al SCSI.

De todos modos, aunque SCSI es superior, se empieza a considerar la alternativa SATA para sistemas informáticos de gama alta ya que su rendimiento, aunque inferior, no lo es demasiado y su diferencia de precio resulta bastante ventajosa. Además, Serial ATA también permite que cada disco duro trabaje sin interferir a los demás.

El bus USB

El bus USB (acrónimo del inglés *Universal Serial Bus*, bus universal en serie, en castellano) es una propuesta de un nuevo estándar para la conexión, originalmente, de periféricos de baja velocidad (en su primera especificación permitía alcanzar tan sólo 1,5 Mbps. –obsérvese que la "b" minúscula significa bits; esta velocidad en bytes supone apenas 192 KBps.) tales como teclados, ratones y *joysticks*. El bus tiene conexiones en forma de puertos que permiten conectar periféricos a una computadora. Es un bus de (relativamente) reciente aparición, pues fue creado en 1996 por un consorcio formado por siete empresas: IBM, Intel, Northern Telecom, Compaq, Microsoft, Digital Equipment Corporation y NEC.

El bus USB permite la conexión "en caliente" (*hotplug*) de los dispositivos, es decir, sin necesidad de tener que reiniciar el ordenador para que el sistema pueda reconocerlos, y soporta la conexión de varios dispositivos simultáneamente: en principio uno en cada puerto. Téngase en cuenta que el diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en las ranuras de expansión de los buses ISA (vigente por aquella época) o PCI, y mejorar las capacidades *Plug & Play* permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Cuando se conecta un nuevo dispositivo, el servidor del controlador USB lo enumera y permite que el sistema operativo "active" el *software* necesario para que pueda funcionar.



Figura 4.36. Un *hub* ("ladrón") USB que permite la conexión de varios dispositivos USB al único puerto al que se conecta

Con USB los periféricos se pueden conectar unos a otros formando una cadena, aunque lo más habitual es que varios dispositivos se conecten a un *hub*. Un *hub* USB permite ampliar fácilmente la capacidad de conexión de dispositivos USB (hasta 256 dispositivos). Sin embargo, por limitaciones del protocolo de comunicaciones que rige el bus, no se pueden tener más de 128 periféricos conectados simultáneamente a la cadena USB.

El estándar USB incluye la transmisión de energía eléctrica y eso supone una ventaja: puede suministrar corriente eléctrica a los periféricos que se le conectan. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra, pero otros gastan tanta energía que necesitan su propia fuente de alimentación.

USB es bastante más rápido que los puertos serie o paralelo, tradicionalmente presentes en los ordenadores personales y para los que el USB supone la alternativa definitiva. Como hemos comentado al principio de esta sección, la versión 1.0 del

USB permitía 1,5 Mbps. y estaba pensada para dispositivos de baja velocidad, como teclados y ratones. Poco después, apareció el estándar USB 1.1, el cual tiene 2 velocidades de transferencia: 1,5 Mbps. para teclados, ratón, joysticks, etc., y *velocidad completa* a 12 Mbps. (1,5 MBps.). Muchos dispositivos fabricados en los años 2000, 2001 y 2002 trabajaban a esta velocidad (por ejemplo, los primeros discos flash USB y muchas impresoras). El problema del USB, no obstante, era que esta baja velocidad impedía la conexión de periféricos externos con una alta demanda de velocidad de transferencia, tales como discos duros externos y cámaras digitales de vídeo[17].

Por ello, surgió hace uno años la versión 2.0. La mayor ventaja del estándar USB 2.0 es añadir un modo de alta velocidad de 480 Mbps. (60 MBps.). Con esta velocidad de transferencia ya se hace posible la conexión de discos duros y cámaras digitales de vídeo. USB 2.0 es compatible hacia atrás con dispositivos USB 1.1. "Hacia atrás" quiere decir que, si hay varios dispositivos conectados a un mismo puerto USB, el puerto funciona a la velocidad del que menos tenga: esto quiere decir que si hubiese conectado un dispositivo 1.1 el bus no funcionaría a más de 12 Mbps. USB 2.0 compite directamente con FireWire, aunque este último es más fiable, veloz y tecnológicamente superior.

La respuesta definitiva a FireWire puede llegar de la mano de la revisión 3.0 del USB (en fase experimental en el momento de elaborar estos apuntes). La especificación definitiva del estándar se ha publicado muy recientemente. Las velocidades de los buses serán 10 veces más rápidas que la de USB 2.0, debido a la inclusión de un enlace de fibra óptica que trabaja con los conectores tradicionales de cobre. Se espera que los productos fabricados con esta tecnología lleguen al consumidor a partir de 2010. En principio, se supone que será compatible con USB 2.0 y tendrá una velocidad en torno a 4,8 Gbps., es decir, del orden de los 600 MBps.

Con USB, se puede conectar un sinfín de periféricos. Por ejemplo: ratones, teclados, escáneres, teléfonos móviles, reproductores multimedia, módems, impresoras, discos duros externos, tarjetas de memoria Compact Flash (o similares), cámaras digitales de fotos (lo que permite que el usuario de ordenadores entre en el mundo de la fotografía digital y pueda crear sus propios álbumes en CD-ROM), cámaras digitales de vídeo (lo que permite la edición de vídeo por parte del usuario de ordenadores)... Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha desplazado a los puertos paralelos porque el USB hace sencillo el poder agregar más de una impresora a un ordenador personal.

Si bien USB no ha remplazado completamente a los teclados y ratones PS/2, hoy en día todas las placas base de PC traen uno o más puertos USB. En el caso de los discos duros, el USB es poco probable que reemplace completamente a los buses como el ATA (IDE) y el SCSI porque el USB tiene un rendimiento un poco más lento que esos otros estándares. El nuevo estándar Serial ATA permite tasas de transferencia de hasta 300 MBps. Sin embargo, el USB tiene una importante ventaja en su habilidad de poder instalar y desinstalar dispositivos sin tener que abrir el sistema, lo cual es útil para dispositivos de almacenamiento externo. Hoy en día, una gran parte de los fabricantes ofrece dispositivos USB portátiles que ofrecen un rendimiento casi indistinguible en

[17]Estos dispositivos se conectaban al bus FireWire, de mucha mayor velocidad.

comparación con los IDE/ATA. Sin embargo, la inminente llegada del USB 3.0 abre algunas incógnitas al respecto, ya que la velocidad anunciada supera con creces la de SATA-2 (que es de 300 MBps.). También hay que destacar el hecho de que, continuando con esta especie de "escalada armamentística", la esperada nueva versión de SATA (SATA-3) permitirá duplicar las velocidades de SATA-2, superando de esta manera los 600 MBps.

El bus FireWire (IEEE 1394)

El bus IEEE 1394, popularmente conocido como *FireWire* (Apple Inc.) o también como *i.Link* (Sony) es un estándar multiplataforma para entrada/salida de datos en serie a gran velocidad. Suele utilizarse, principalmente, para la interconexión con ordenadores personales de dispositivos digitales externos que requieren elevadas tasas de transferencia, tales como cámaras digitales de vídeo.

FireWire es un bus de alta velocidad, desarrollado por *Apple Computer* a mediados de los noventa. A principios de este siglo fue adoptado por los fabricantes de periféricos digitales hasta convertirse en un estándar establecido: el estándar multiplataforma IEEE 1394. Su elevada tasa de transferencia de datos permite, por ejemplo, la captura directa de imágenes desde cámaras digitales de vídeo en el ordenador sin necesidad de convertirlas y sin pérdidas. En la actualidad, rivaliza en prestaciones con el estándar USB 2.0.

Sony utiliza el estándar IEEE 1394 bajo la denominación *i.Link*. Éste implementa fielmente el estándar, pero sólo utiliza 4 conexiones de las 6 disponibles en la norma, suprimiendo las dos conexiones encargadas de proporcionar energía al dispositivo, que, en consecuencia, tendrá que proveerse de ella mediante una toma separada.



Figura 4.37. Un puerto FireWire en un portátil

Entre las características más sobresalientes de FireWire podemos mencionar su velocidad, 400 Mbps.(en torno a los 50 MBps.), y la posibilidad de conectar hasta 63 dispositivos distintos a un mismo bus, con una longitud máxima del cable de 4,5 metros. FireWire también permite conexión en caliente (*hotplug*), al igual que USB. FireWire soporta la conexión interna de dispositivos y, de manera similar a USB, proporciona alimentación eléctrica por el bus. En este sentido, proporciona mayor potencia que el USB: los dispositivos con FireWire pueden proporcionar o consumir hasta 45 vatios (USB 2.0 sólo proporciona 2,5 W), más que suficiente para discos duros de alto rendimiento y baterías de carga rápida.

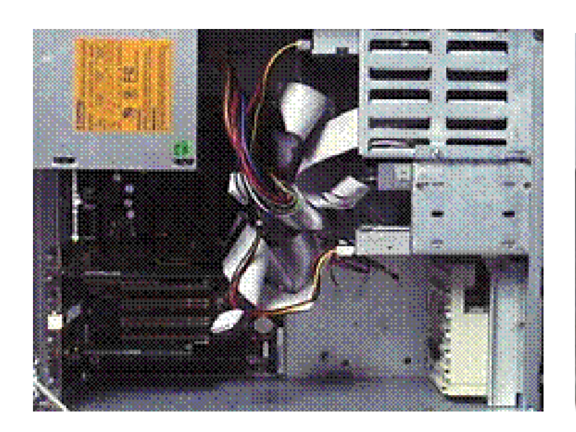
Los ordenadores de Apple, los populares Macintosh, lo incluían ya de serie prácticamente desde su creación. En los ordenadores personales (PC) ha sido más difícil de encontrar hasta hace poco: hoy en día muchas de las placas base para PC (y sobre todo los portátiles) incluyen un puerto IEEE 1394. FireWire también puede incorporarse a un PC mediante una tarjeta de expansión (para PCI). Habitualmente, al bus FireWire se le pueden conectar cámaras digitales de vídeo, regrabadoras DVD-ROM, discos duros

externos, etc.

Recientemente, se ha desarrollado una nueva especificación, *FireWire 800* (norma IEEE 1394b, popularmente conocida como FireWire 2), que tiene una tasa de transferencia de 800 Mbps. (100 MBps.) y que es compatible con dispositivos FireWire 400 (funcionando a menor velocidad).

Serial ATA

El bus Serial ATA o SATA (acrónimo de *Serial Advanced Technology Attachment*) es una interfaz para transferencia de datos entre la placa base y algunos dispositivos de almacenamiento, como pueden ser los discos duros, u otros dispositivos de altas prestaciones. SATA está sustituyendo a la conexión IDE/EIDE/ATA tradicional (también denominada *Parallel ATA* o PATA). En general, SATA proporciona mayores velocidades de transferencia y, por tanto, mayor productividad del disco, ofrece un mejor aprovechamiento cuando hay varios discos (al no tener la limitación del esquema de conexión maestro/esclavo típico de IDE, ya que la conexión es punto a punto), permite una mayor longitud del cable de transmisión de datos y tiene la capacidad de conectar discos en caliente (con el ordenador encendido).





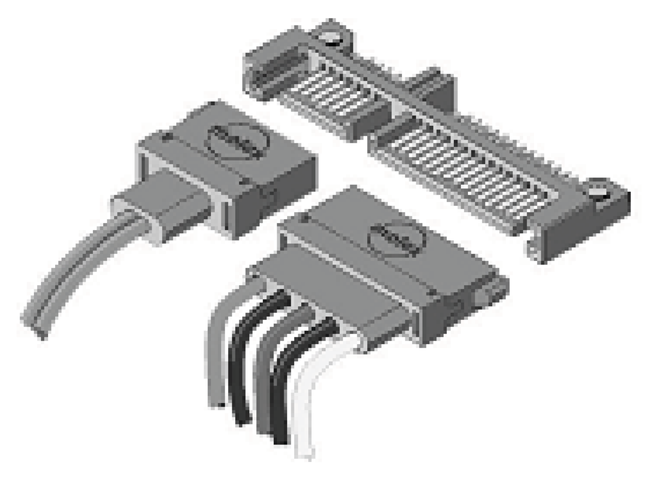


Figura 4.38. Bus *Serial ATA* en el PC. La ilustración de abajo permite comparar la conexión SATA con la conexión del cable tradicional Parallel ATA. Las otras dos imágenes permiten comparar el cableado empleado para la conexión de un disco duro interno (SATA es mucho más fino)

Actualmente es una interfaz extensamente aceptada y estandarizada en las placas base que se fabrican para ordenadores personales. La Organización Internacional SATA (SATA-IO) es el grupo responsable de desarrollar, de manejar y de conducir la adopción de especificaciones estandarizadas para SATA. Entre las principales características de SATA destacan las siguientes:

- Los dispositivos conectados a la controladora no comparten cable, ya que se conectan directamente, pues se trata de una conexión punto a punto, mediante *cable fino* (véase la Figura 4.38).
- La longitud máxima del cable de conexión desde la controladora al dispositivo puede ser de hasta 1 metro, frente a los 45 cm. que, como mucho, permite el estándar *Parallel ATA*.
- Al ser una conexión punto a punto entre dispositivo y controladora, no hace falta terminación ni un esquema de conexión maestro/esclavo como en el PATA tradicional.
- Como se utiliza cable fino, conseguimos una mejor ventilación y maniobrabilidad en el interior del PC (los cables PATA ocupan mucho espacio y resultan molestos).
- SATA proporciona mayor velocidad de transferencia de datos: 150 MBps. frente a los 133 MBps de los IDE actuales. La segunda generación SATA ya permite duplicar esta velocidad, alcanzando los 300 MBps.
- En principio, soporta prácticamente todos los tipos de dispositivos que se pueden conectar a IDE (fundamentalmente discos duros y regrabadoras DVD).
- Permite "conexión en caliente" (*hotplug*).
- Las placas base actuales para PC ya incluyen este tipo de "interfaz" (bus). Muchos portátiles de gama media/alta incorporan este tipo de bus en lugar del ATA tradicional.

4.4. PERIFÉRICOS

Se denominan periféricos a las unidades o dispositivos a través de los cuales la computadora se comunica con el mundo exterior. Así pues un periférico es un dispositivo que, sin pertenecer al núcleo fundamental de la computadora, la CPU y la memoria central, permite realizar operaciones de entrada/salida (E/S) complementarias al proceso de datos que realiza la CPU. En general, son dispositivos que conectamos directa o indirectamente (a través de puertos, ranuras o conexiones) a la placa base. Los periféricos se pueden clasificar en 4 grupos, atendiendo a su funcionalidad. Así pues, podemos distinguir:

- De entrada (por ejemplo, el teclado y el ratón).
- De salida (por ejemplo, la pantalla o monitor).

- De entrada/salida (por ejemplo, un módem).
- De almacenamiento secundario (por ejemplo, los discos duros).

Además, según si se conectan y ubican dentro de la caja o bien desde fuera de ella, los periféricos pueden ser *internos* (como discos duros, lectores y regrabadores de CD-ROM o DVD y tarjetas de expansión) o bien *externos*, que pueden conectarse a conexiones de las tarjetas de expansión (como altavoces, monitores, micrófonos, etc.) o a puertos ya integrados en la placa base (como teclados, ratones, dispositivos USB, impresoras, etc.). A continuación, vamos a estudiar algunos de los periféricos más utilizados en la actualidad.

Sobre los dispositivos periféricos...

Generalmente, en un dispositivo periférico se distinguen dos partes: una parte mecánica, el dispositivo "en sí", y una parte electrónica que lo comunica con el bus correspondiente de la placa base, el controlador.

Desde el punto de vista del *software*, necesitaremos un *driver* o manejador (programa) que el sistema operativo pueda emplear para interactuar con el controlador (parte electrónica) y así poder utilizar el dispositivo. Realmente es el controlador el que se ocupa de manejar la parte mecánica.

4.4.1. El monitor

El monitor es el principal medio de salida de información desde el ordenador, ya que permite representar los datos de tipo texto o gráficos procesados por éste. Él monitor se limita a representar la imagen que le envía la tarjeta de vídeo (también conocida como tarjeta gráfica). Es en ésta donde sucede la mayor parte del "trabajo". Ambos dispositivos deben estar compensados: no merece la pena que la tarjeta pueda trabajar con resoluciones de 1.600×1.200 si el monitor no las soporta. En los monitores modernos la imagen se forma mediante puntos (píxeles) y las imágenes se representan mediante alguno de los modelos de color explicados en el capítulo 3 ("Codificación de la información"). El monitor es uno de los componentes en los que más merece la pena fijarse al comprar un ordenador nuevo, ya que sus prestaciones afectan a nuestra vista. Hasta hace unos pocos años, podíamos encontrarnos con dos tipos de monitores:

- De rayos catódicos. Funciona de manera similar a un televisor convencional, pero con una imagen de mayor calidad: un haz de rayos catódicos incide en la pantalla (que es el ánodo). Ésta está recubierta de un material que se ilumina al recibir el haz, permaneciendo iluminado un corto tiempo. El haz recorre la pantalla dibujando los puntos línea a línea. Si en cada recorrido se dibujan todas las líneas, el monitor es *no entrelazado* (lo habitual). Los monitores *entrelazados* dibujaban una línea sí, otra no y viceversa (primero las pares y luego las impares). Un ejemplo: un monitor *no entrelazado* con una frecuencia de refresco

de 70 Hz, renueva completamente la imagen en pantalla 70 veces por segundo, mientras que un monitor *entrelazado* sólo la renovaría 35.

- De TFT. Emplean tecnología de plasma. Son pantallas planas, poco profundas. La imagen causa menos fatiga visual. En la actualidad, su precio ya es asequible. Los ordenadores portátiles fueron los primeros en incorporar pantalla TFT: ahora son más baratos debido en parte al abaratamiento de este tipo de pantallas. Hoy en día también es lo estándar para equipos de sobremesa.



Figura 4.39. A la izquierda, un monitor de pantalla plana (TFT). A la derecha, un monitor de rayos catódicos

Actualmente la mayoría de monitores utilizan la tecnología de plasma (TFT). Algunas de las principales características del monitor son:

Tamaño medido en pulgadas. Lo que se mide es la longitud de la diagonal de la pantalla. Hoy en día los monitores más frecuentes son de 17 ó 19 pulgadas. Sin embargo, existen monitores de gama alta de 21, 22 y más.

Frecuencia de barrido vertical o velocidad de refresco. Indica el número de veces que se renueva la imagen por segundo. Se expresa en Hz. A mayor velocidad, menor fatiga visual. En los (viejos) monitores de rayos catódicos, se aprecia parpadeo en la imagen por debajo de los 75 Hz.

Frecuencia de barrido horizontal. Se calcula como la resolución vertical multiplicada por la frecuencia de refresco.

Máxima resolución. Número de píxeles (puntos de la imagen) que soporta como máximo tanto en ancho como en alto.

Dot pitch. Es la separación entre las perforaciones de la rejilla de potencial, esto es, el tamaño de los puntos de la pantalla (**no** confundir con los píxeles). En el caso de los monitores de rayos catódicos dependía del tubo de imagen. En los TFT depende de la tecnología empleada. A menor *dot pitch*, mejor calidad de imagen.

Además, cabe tener en cuenta otro tipo de consideraciones. Por ejemplo, es importante que el cable sea de 15 conectores (hoy en día prácticamente todos lo son, ya que es un estándar que ya fue marcado en su día por IBM con su VGA). Si el cable está separado, la reparación es más sencilla. También, el monitor se puede intentar "forzar" para obtener frecuencias de refresco más altas. Sin embargo se puede (en teoría) dañar. Por lo tanto, si se desea intentarlo es conveniente consultar el manual.

4.4.2. Las tarjetas de expansión

Estas tarjetas se conectan en las ranuras de expansión que ofrecen ciertos buses de la placa base (ISA, PCI, PCI Express o AGP) para permitir aumentar las capacidades del ordenador. En la mayoría de los casos son controladoras que permiten la conexión de otros dispositivos y que pueden manipularlos (por ejemplo, una tarjeta gráfica permite la conexión, entre otros, de un monitor que mostrará la imagen que le "envía" la propia tarjeta). Podemos encontrarnos tarjetas de todo tipo para ordenadores personales. Las más habituales son:

- La tarjeta de vídeo, popularmente conocida como tarjeta gráfica. Es absolutamente necesaria para el funcionamiento del PC, incluso en "modo texto". Hoy en día incluyen capacidades de "aceleración 3D" (procesador específico necesario para el correcto funcionamiento de aplicaciones CAD –diseño asistido por ordenador– y juegos de última generación, entre otros). El monitor se conecta directamente a la tarjeta de vídeo para poder mostrar así las imágenes.
- La tarjeta de sonido. Es imprescindible si se desea que el ordenador emita (escuchar música, por ejemplo) y grabe sonido (a través de un micro, por ejemplo). Los altavoces se conectan a la tarjeta de sonido.
- Tarjeta de red *ethernet*. Necesitamos esta tarjeta si queremos conectarnos directamente a una red Ethernet (como la de la UJI). En la actualidad, también se emplea como interfaz de conexión a una red de cable a través de un cable-módem o (con menor frecuencia) a una red telefónica de tipo ADSL a través de un *router* específico.

Las tarjetas y demás dispositivos interactúan con el sistema operativo del ordenador, que es quien se encarga de manipularlas mediante la utilización de programas específicos (*drivers*) proporcionados por el fabricante o bien incluidos como módulos específicos en el sistema. Esta interacción se realiza habitualmente:

- ► **Mediante interrupciones**. El mecanismo de interrupciones permite enviar señales hardware (de entre una batería que vienen predeterminadas en la circuitería del ordenador) que el sistema operativo (S.O.) está preparado para "capturar". Cada una de estas interrupciones está asociada a un dispositivo concreto. De esta forma, cuando un dispositivo "llama al timbre", es decir, emite la señal correspondiente a su interrupción, el sistema operativo ya sabe quién está llamando. Las tarjetas hacen peticiones cuando tienen información que el S.O. debe tratar. El ordenador tiene una tabla de interrupciones IRQ (*Interrupt Requests*) que asocia a los diferentes dispositivos. Entre las más comunes, destacan:

IRQ	uso
0	temporizador
1	teclado
3	puerto serie COM2
4	puerto serie COM1
5	puerto paralelo LPT2
6	controlador de disquetera
7	puerto paralelo LPT1
14	controlador disco duro

- **Mediante puertos de entrada/salida**. Los puertos de entrada/salida permiten pasar información complementaria en la comunicación: por ejemplo, para que el sistema indique al dispositivo la dirección de memoria donde están los datos que debe procesar.
- **Mediante acceso directo a memoria** (popularmente conocido como DMA, del inglés *Direct Memory Access*). Cuando la tarjeta o el dispositivo dispone de datos, los almacena directamente en memoria RAM para que la CPU pueda acceder a ellos más rápidamente. De esta manera, aumentamos la velocidad de proceso de datos. Tanto el dispositivo como la CPU dejan los datos directamente en una zona de la memoria RAM habilitada a tal efecto. Por tanto, se evitan las esperas en el bus.

Cuando queramos instalar una tarjeta de expansión en nuestro ordenador tenemos que tener en cuenta el tipo de bus al que la vamos a conectar, la disponibilidad de ranuras (*slots*) libres en dicho bus para poder conectarla y si las tarjetas son configurables automática o manualmente (normalmente, lo serán automáticamente, mediante *Plug & Play*).

Si son configurables automáticamente, a priori no debería haber problemas ya que se usan las capacidades *Plug & Play* que ofrece, por ejemplo, el bus PCI y la BIOS del ordenador. La información acerca de la configuración de la tarjeta pasa al sistema operativo y, entonces, basta con instalar el *driver* (programa manejador) correspondiente para que ésta pueda funcionar. En cambio, si son configurables manualmente hay que buscar una IRQ y un rango de direcciones libres (para entrada/salida). Después hay que configurar la tarjeta, bien mediante la manipulación de sus *jumpers* (conectores), bien con algún programa que proporcione el fabricante. En la actualidad, la mayoría de las tarjetas y dispositivos que se fabrican para ordenadores personales son *Plug & Play*, por lo que rara vez tendremos que preocuparnos por estos menesteres.

Si no tenemos cuidado con la configuración manual de una tarjeta (que normalmente será antigua), es posible que al instalarla se produzca una *colisión* o conflicto entre interrupciones asignadas (IRQ), direcciones de puertos de entrada/salida o direcciones de memoria (para el DMA). Es decir, es posible que ocurra alguna de las siguientes situaciones:

- Que dos tarjetas tengan asignado el mismo número IRQ.
- Que dos tarjetas tengan asignados los mismos puertos de entrada/salida.
- Que dos tarjetas traten de escribir en las mismas posiciones de memoria RAM.

Esta situación también puede ocurrir incluso con tarjetas *Plug & Play* (PnP), de configuración automática. Como ya comentamos en la sección 4.3.5, en algunas ocasiones (sobre todo si se trata de tarjetas PnP) esto no supondrá mayor problema ya que hay algunas tarjetas que están preparadas para compartir una misma interrupción u otros recursos y, de alguna forma, la situación es gestionada correctamente por los dispositivos implicados. Pero cuando ocurre el conflicto y alguna de las tarjetas no funciona (porque está "tapada" por la otra), hay que resolverlo:

- Modificando la asignación de IRQ mediante la manipulación de los *jumpers* (conectores o puentes) de la tarjeta.
- Configurando adecuadamente las tarjetas en la BIOS (en el arranque del ordenador) mediante el cambio de asignación automática a manual o bien desde el panel de control de Windows (o con las herramientas de configuración de Linux).

En cualquier caso, ambas son operaciones delicadas que requieren un buen conocimiento de la situación y saber muy bien lo que se está haciendo (¡hay que leerse el manual!). Como podemos apreciar, la solución puede ser difícil, aunque tal vez lo más cómodo en estos casos, si se dispone de ranuras libres en el bus PCI de la placa, es cambiar a otra ranura uno de los positivos que entran en conflicto. Al arrancar de nuevo el ordenador, se reasignan las interrupciones y, con suerte, el problema se resolverá solo. Hace unos años, esta sencilla táctica le sirvió al que suscribe para solucionar un conflicto entre una tarjeta Ethernet y otra de sonido (ambas PCI) en un PC (con procesador AMD Athlon XP a $1{,}4$ GHZ) equipado con Linux.

La tarjeta de vídeo

La tarjeta gráfica es la que se encarga de procesar la información que le envía la CPU y, a su vez, de transmitirla al monitor. Mantiene la imagen que se ha de mostrar en pantalla almacenada en su propia memoria.

Hoy en día las tarjetas gráficas se fabrican para ranuras PCI Express (el último "estándar"), fundamentalmente, y AGP (a extinguir). Las tarjetas, generalmente, tienen algún tipo de aceleración (procesador específico). No se deben confundir con las tarjetas de televisión (para ver televisión en el ordenador) o las de vídeo, a las cuales se les conectan cámaras para enviar o grabar vídeo.

La tarjeta gráfica es la que tiene una conexión de salida para que pueda conectarse el monitor, y así, éste pueda mostrar lo que ocurre en el computador. Muchas incorporan un puerto de televisión (para conectarlas a un monitor de televisión) e incluso salida de vídeo HDMI (acrónimo del inglés *High-Definition Multimedia Interface*), una norma de audio y vídeo digital cifrado sin compresión apoyada por la industria para que sea el sustituto del euroconector (pensando en la conexión con televisores digitales, pero no sólo eso).

¿Sabías que…?

Históricamente se definieron una serie de estándares para trabajar con las tarjetas gráficas: MDA, CGA, Hércules, EGA, VGA. En la actualidad están obsoletos.

Tan sólo sobrevive el estándar VGA, que soportan todas las tarjetas mediante las extensiones Súper VGA (que no son en sí un estándar) y las especificaciones VESA (que actúan como un estándar *de facto*).

La imagen que muestra el monitor y que almacena la tarjeta gráfica en su memoria es una matriz de píxeles. Como ya sabemos (lo estudiamos en el capítulo 3, "Codificación de la información"), un píxel es un punto que puede presentar diferentes colores. También vimos que los colores se forman "mezclando" tonos de tres colores básicos: rojo, verde y azul (según el modelo de color RGB, del inglés *Red, Green, Blue*; éste es el modelo de color más usado), aunque hay otros modelos de color. Si una imagen es en blanco y negro, es posible representar cada píxel con un bit; si tiene 256 niveles de gris, cada píxel es un byte; si la imagen tiene 256 tonos por cada color básico (RGB), cada píxel son 3 bytes (24 bits). Con 24 bits por píxel es posible representar más de 16 millones de colores.

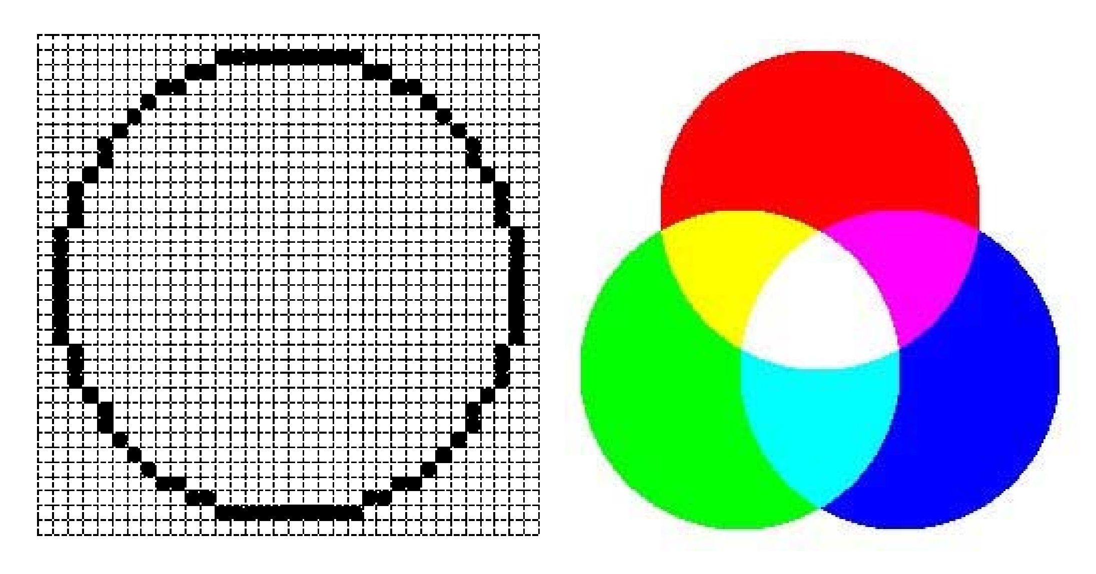


Figura 4.40. A la izquierda, un mapa de bits. A la derecha, el modelo de color RGB, que permite formar cualquier color mediante la superposición de sus tres colores básicos (rojo, verde y azul)

En resumen, las tarjetas gráficas incorporan su propia memoria para gráficos, ya que, básicamente, permiten visualizar en un monitor una zona de memoria en la que hay una imagen almacenada como una matriz de píxeles. Precisamente, el tamaño de esta matriz de píxeles determina la resolución de la pantalla gráfica con la que estamos trabajando. Obviamente, las características del monitor (tamaño, velocidad de refresco) influyen en las resoluciones que podemos fijar. Para hacernos una idea, veamos algunas resoluciones típicas y la memoria que consumen (aproximadamente) en función del número de tonos/colores que empleemos (según modelo de color):

	256 (1 byte)	65.536 (2 bytes)	16.777.216 (3 bytes)
640×480	512 KB	1 MB	1 MB
800×600	512 KB	1 MB	1,5 MB
1.024×768	1 MB	1,5 MB	2,5 MB
1.280×1.024	1,5 MB	2,5 MB	4 MB
1.600×1.200	2 MB	4 MB	5,5 MB



Figura 4.41. Una tarjeta gráfica con procesador 3D de la marca nVIDIA

En el momento de redactar estos apuntes, lo "estándar" en cuanto a equipamiento de memoria de una tarjeta gráfica serían 256 MB de memoria propia (no compartida con la RAM del ordenador). A la vista de la tabla anterior, cabe preguntarse "¿para qué necesitan las tarjetas gráficas tanta memoria si, por ejemplo, con 8 ó 10 MB sería suficiente para el trabajo cotidiano?".

La respuesta a esta pregunta es que es precisamente *para agilizar el trabajo con gráficos* en el ordenador. Hoy en día, tener más memoria en la tarjeta gráfica no sólo implica la posibilidad de trabajar a mayores resoluciones y poder representar mayor número de colores (a igual resolución), sino que también incrementa la velocidad de las operaciones con gráficos (aceleración 3D). La memoria propia de la tarjeta es otro factor importante para el rendimiento de las aplicaciones con uso intensivo 3D (el otro es el propio procesador de la tarjeta). En conclusión, cuanta más memoria tenga la tarjeta, mejor, pero ésta también será más cara entonces.

Hay que tener en cuenta que los fabricantes, con el objetivo de colocar en el mercado tarjetas más baratas, venden tarjetas con *memoria compartida*. Eso quiere decir que no tienen (o tienen poca) memoria propia y que, por tanto, usan la memoria RAM del ordenador. Este tipo de tarjetas son más baratas, pero hay que considerar que las prestaciones son peores (especialmente en 3D: *rendering* y CAD, por ejemplo).

Llegado este punto, conviene comentar algunas cosas más respecto al concepto de "aceleración" de la tarjeta gráfica que acabamos de mencionar. Pero antes, permítasenos poner un ejemplo que servirá para ilustrar al lector lo que significa trabajar con gráficos en un ordenador.

Imaginemos que estamos trabajando en un entorno gráfico de usuario tal como el escritorio KDE o el de Windows XP. La simple operación de desplazar una ventana de 430×300 por un escritorio de 1.280×1.024 con millones colores (24 bits de profundidad de color) implica mover un bloque de memoria de 378 KB. Además, esto debe hacerse con la mayor velocidad y suavidad posibles para evitar efectos "indeseables" (por ejemplo, que la ventana deje un "rastro" de su paso por el escritorio). Si es el procesador (CPU) del ordenador quien ha de encargarse de estas operaciones, el sistema tenderá a ralentizarse (imaginemos cuántas operaciones de este tipo podemos llegar a realizar en un entorno gráfico de usuario en relativamente poco tiempo).

Ocurre lo mismo, pero incluso más agravado, a la hora de trabajar con programas en tres dimensiones, tales como aplicaciones de CAD (diseño asistido por ordenador), modelado sólido (construcción de modelos de objetos en 3D) o juegos de última generación, que emplean de forma intensiva operaciones matemáticas para gestionar polígonos, moverlos, visualizarlos, aplicar texturas, etc. Este tipo de aplicaciones requieren la visualización en tiempo real de gráficos 3D.

Dado que las aplicaciones exigen cada vez más prestaciones desde el punto de vista gráfico, es deseable liberar al procesador principal (CPU) de parte de la carga que representa el dibujar en la pantalla. Para liberar al procesador de los cálculos necesarios, las tarjetas gráficas incorporan *circuitos especializados* en visualización 3D que implementan las funciones más utilizadas por los diferentes programas y juegos. Son los procesadores de la tarjeta de vídeo o *aceleradores gráficos* 3D (véase la Figura 4.41). Algunas tarjetas 3D de última generación incorporan, incluso, procesadores más potentes que algunos micros para PC (por ejemplo, que el Pentium 4).

Completando información

Al principio, los aceleradores se empleaban para tareas típicas del entorno de ventanas (por ejemplo, mover bloques de un sitio a otro de la pantalla). Actualmente asumen la responsabilidad de manejar los gráficos en dos (y tres) dimensiones, liberando así a la CPU para otros cometidos.

Las tarjetas con aceleración 3D más modernas incluyen, por *hardware*, rutinas para el mapeado de texturas, manipulación de polígonos y vectores, etc.

La aceleración 3D se utiliza para procesar la gran cantidad de operaciones necesarias para representar en dos dimensiones una escena tridimensional. Es especialmente necesaria en aplicaciones como juegos y diseño en 3D (programas de modelado, animación, *rendering* o dibujo de perspectivas, CAD), donde es normal tener que hacer del orden de 75 dibujos (imágenes), o más, por segundo (fps. del inglés *frames per second*).

Con el fin de que los programadores tuviesen mayor facilidad para desarrollar aplicaciones que aprovechasen el juego específico de instrucciones de los procesadores 3D de las tarjetas gráficas, se crearon una serie de *bibliotecas de programación*, API (del inglés *Application Programming Interface*, Interfaz de Programación de Aplicaciones, en castellano). Un API es, básicamente, el conjunto de funciones y procedimientos (o métodos, si se refiere a programación orientada a objetos) que ofrece(n) cierta(s) biblioteca(s) para ser utilizado por otros programas como una capa de abstracción.

Estas API pretendían unificar, bajo una interfaz común, las diferentes funciones que habían de usar los programas para “comunicarse” con la tarjeta gráfica, y así realizar una abstracción del *hardware*. Inicialmente, prácticamente cada fabricante utilizaba su propia interfaz. Actualmente sólo sobreviven, mayoritariamente, dos:

Direct3D. Es parte de DirectX, que es una colección de API creadas para facilitar las complejas tareas relacionadas con la programación de juegos en la plataforma

Windows de Microsoft. Direct3D es una de estas API, propiedad de Microsoft, y disponible tanto en los sistemas Windows de 32 y 64 bits, como en sus consolas Xbox y Xbox 360, orientada exclusivamente a la programación de gráficos 3D.

El objetivo de Direct3D es facilitar el manejo y trazado de entidades gráficas elementales, tales como líneas, polígonos y texturas, en cualquier aplicación que despliegue gráficos en 3D, así como efectuar de forma transparente transformaciones geométricas sobre dichas entidades. Direct3D proporciona también una interfaz transparente con el *hardware* de aceleración gráfica común a todas las tarjetas que la soportan. Se usa principalmente en aplicaciones donde el rendimiento 3D es fundamental, aprovechando el *hardware* de aceleración disponible en la tarjeta gráfica.

La última versión de DirectX, y por tanto de Direct3D, es la 10.0. Esta versión sólo funciona con el sistema operativo Windows Vista, mientras que DirectX 9.0 trabaja con toda la familia de Windows a partir de Windows 98. Sólo las tarjetas gráficas recientes comienzan a soportar la versión 10.0 de DirectX.

OpenGL. OpenGL es una especificación estándar, es decir, un documento que describe un conjunto de funciones y su comportamiento exacto, que define un API multilenguaje y multiplataforma para escribir aplicaciones que produzcan gráficos 2D y 3D. Fue desarrollada inicialmente por la empresa *Silicon Graphics Inc.* (SGI) a principios de los noventa. Su nombre viene del inglés *Open Graphics Library*, cuya traducción sería algo así como "biblioteca de gráficos abierta" (o mejor, libre, teniendo en cuenta su política de licencias).

A partir de la especificación OpenGL, los fabricantes de tarjetas gráficas crean implementaciones, esto es, bibliotecas de funciones (código real) creadas para enlazar con las funciones de la especificación OpenGL, utilizando aceleración hardware cuando sea posible. Dichos fabricantes tienen que superar pruebas específicas que les permitan calificar su implementación como una implementación de OpenGL (de ahí la etiqueta de *OpenGL certified* que llevan algunas gráficas).

OpenGL se utiliza en campos como CAD, representación científica de información, realidad virtual, simulación de vuelo o desarrollo de videojuegos, en el que su principal competidor es el API Direct3D de Microsoft.

Algunas curiosidades

En su momento, existió otro API 3D muy popular, que rivalizaba con Direct3D y OpenGL: se trataba de *Glide*. Éste era propio de las tarjetas con procesador gráfico 3D *Voodoo Graphics* de la empresa *3dfx Interactive*.

Glide consiguió imponerse efímeramente (años 1996, 1997 y 1998) a los demás, gracias a la aceptación de estos procesadores 3D por su elevado rendimiento en comparación con otras soluciones.

continúa en la página siguiente ...

. . . viene de la página anterior

Sin embargo, la mejora de los otros API de desarrollo, Direct3D y OpenGL, y la constante mejora en prestaciones de las tarjetas gráficas de sus rivales, especialmente nVIDIA con su TNT y ATI con el chip Rage, harían que Glide quedara obsoleto (año 1999). A finales del año 2000, la empresa 3dfx desapareció del mercado, siendo absorbida por nVIDIA Corporation, uno de sus rivales.

Glide era un API cerrado (propiedad de 3dfx) pensado para que los desarrolladores de juegos lo utilizaran al programar juegos 3D.

Se trataba, en esencia, de un pequeño subconjunto de OpenGL implementado en el juego de instrucciones del procesador 3D. Glide proporcionaba acceso directo al *hardware* interno de las tarjetas *Voodoo* a los programadores. Esta estrategia difería de la seguida por otros API 3D de la época (Direct3D, OpenGL y QuickDraw 3D), los cuales ocultaban el acceso de bajo nivel al *hardware* tras una capa de abstracción, con el objetivo de proporcionar a los desarrolladores una interfaz estándar independiente del mismo.

DirectX se ha desarrollado exclusivamente para la plataforma Windows. A pesar de ello, se encuentra en fase de desarrollo una implementación de este API para sistemas Unix (en particular Linux) conocida como *Cedega*, realizada por la empresa *Transgaming* y orientada a la ejecución bajo sistemas Unix de juegos desarrollados para Windows.

Hace unos años, concretamente en 1997, Microsoft alcanzó un acuerdo con SGI, con el objeto de fusionar totalmente ambas plataformas: DirectX y OpenGL (de hecho, en DirectX 6 ya estaba presente gran parte de la interfaz OpenGL). A este proyecto se le denominó *Fahrenheit* y se trataba de un esfuerzo cooperativo con el objetivo de unificar las interfaces de OpenGL y Direct3D. En 1998 se uniría al proyecto *Hewlett-Packard*. Pese a tener un inicio prometedor, el proyecto fracasó debido a restricciones financieras en SGI y la falta general de apoyo por parte de la industria, y fue finalmente abandonado en 1999. En la actualidad, DirectX y OpenGL caminan por senderos distintos.

La tarjeta de sonido

Una tarjeta de sonido es una tarjeta de expansión que permite la entrada y salida de audio, es decir grabar y reproducir sonido con el ordenador, bajo el control de un programa informático. Su utilización consiste, básicamente, en proporcionar audio a las aplicaciones multimedia.

Las tarjetas de sonido típicas incorporan un chip (procesador digital de la señal o DSP del inglés *Digital Signal Processing*) de audio que, por lo general, contiene el conversor digital-analógico, el cual se encarga de convertir las señales de audio analógicas en digitales y viceversa.

Los principales conectores de una tarjeta de sonido son: salida analógica estéreo (para conexión de altavoces externos o auriculares, por ejemplo), y micrófono y entrada analógica *line-in*. Muchas tarjetas de sonido siguen un estándar *de facto*, imitando el funcionamiento de las tarjetas *SoundBlaster* del fabricante *Creative*. Estas tarjetas fueron muy populares (con una buena relación precio/prestaciones) durante la segunda mitad de los noventa y principios de este siglo. Actualmente, casi todas las tarjetas de sonido se han adaptado al estándar PC99 de Microsoft. Este estándar consiste en asignar un color a cada conector externo:



Figura 4.42. Una tarjeta de sonido SoundBlaster

Color	Función
Rosa	Entrada analógica para micrófono.
Azul	Entrada analógica "Line-In"
Verde	Salida analógica para la señal estéreo principal (altavoces frontales).
Negro	Salida analógica para altavoces traseros.
Plateado	Salida analógica para altavoces laterales.
Naranja	Salida Digital SPDIF.

Queremos mencionar que algunas veces la salida Digital SPDIF es utilizada como salida análoga para altavoces centrales. Se debe tener en cuenta que el PC suele ir equipado con un altavoz interno de baja calidad que sólo se usa para emitir pitidos de aviso. Hoy en día, se han puesto de moda las tarjetas de sonido de gama alta, equipadas con sonido Dolby y que permiten la conexión de equipos de altavoces y ecualizadores 5 y 5.1 para aplicaciones multimedia y lúdicas ("cine en casa"), tal como muestra la Figura 4.43.



Figura 4.43. Un equipo de altavoces 5.1 para su conexión a una tarjeta de sonido con capacidad de sonido envolvente

4.4.3. El disco duro

La memoria RAM pierde toda la información que contiene cuando se apaga el ordenador. Sin embargo, el disco duro es capaz de mantener la información. Gracias a la utilización de un sistema de ficheros por parte del sistema operativo, que se crea en el momento de formatear el disco, los datos se almacenan en el disco duro como ficheros organizados en una jerarquía de directorios.

El disco duro es, normalmente, un dispositivo interno del ordenador. Hoy en día, también se venden mucho los discos duros externos, que pueden conectarse a puertos USB o FireWire, y de gran capacidad, son muy útiles para copias de seguridad y para transportar fácilmente de un sitio a otro grandes volúmenes de datos. Los discos duros son más rápidos y seguros que las unidades de lectura de disquetes. Hoy en día son frecuentes los discos de 200 y 400 GigaBytes. Los tiempos de acceso a la información de los discos duros se miden en milisegundos.

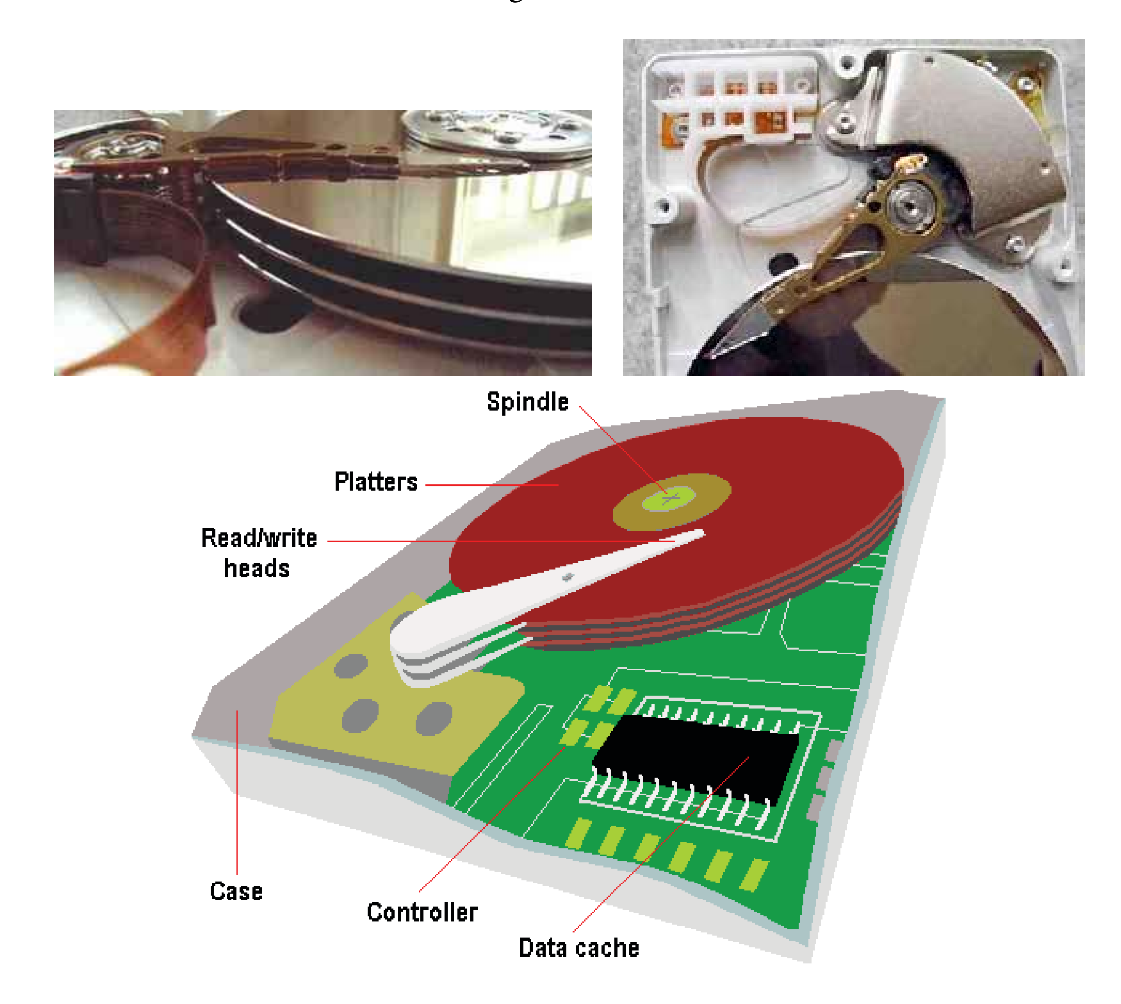


Figura 4.44. Arriba: dos imágenes de los platos y los cabezales de un disco duro. Abajo: un dibujo esquemático que muestra la disposición interna de un disco duro

Los discos duros son el principal medio de almacenamiento secundario del ordenador. En algunos casos se utilizan para apoyar a la memoria (memoria virtual). La parte magnética (el disco propiamente dicho) está encerrado en una cámara estanca libre de polvo. La cabeza de lectura se mantiene separada del disco gracias a un colchón de aire

generado por el giro (desde unas 7.500 hasta más de 10.000 rpm – revoluciones por minuto). En su funcionamiento, el disco no para de girar; aunque esto no es del todo cierto ya que los sistemas, para ahorrar energía y evitar el desgaste, pueden "detener" el disco y "aparcar" los cabezales.

La unidad de disco flexible (disquetera)



Figura 4.45. Una unidad de disco flexible en la que se introduce un disquete

Hasta hace unos años, los ordenadores personales solían llevar preinstalada una unidad de lectura/escritura de disquetes. Este tipo de unidad, actualmente en desuso, permite la utilización de discos flexibles. Estos discos tienen un tamaño de tres pulgadas y media, pero su capacidad de almacenamiento es muy baja (tan sólo "caben" 1,44 MB) y son extremadamente lentos.

Más antiguamente (en los años ochenta), se utilizaban discos de cinco pulgadas y cuarto que tenían, incluso, menor capacidad: 360 KB en doble densidad y 1,2 MB en alta. En la actualidad están totalmente obsoletos, ya que el principal medio de "transporte" de ficheros son las memorias Flash que pueden conectarse a USB (con mucho menos espacio, el de un llavero por ejemplo, podemos transportar cómodamente 8 GB o más).

El funcionamiento es análogo al del disco duro, con la excepción de que la cabeza *sí* toca la superficie del disco. Esto hace que el disquete se desgaste notablemente, por lo que se para el motor de la unidad de disco en los periodos de inactividad. También es motivo de que se estropeen frecuentemente (si han sido muy utilizados): el soporte es poco fiable, aunque, a cambio, tanto la unidad como los disquetes eran muy baratos.

Como curiosidad, queremos indicar que fueron los ordenadores personales de Apple los primeros que dejaron de incorporar la popular disquetera, mientras en los PC todavía se seguía incluyendo mayoritariamente.

Geometría de los discos duros

Un disco está formado por varios platos o discos (superficies circulares) recubiertos de material magnético que giran de *forma continua* alrededor de un eje (véase la Figura 4.44). Cada una de las

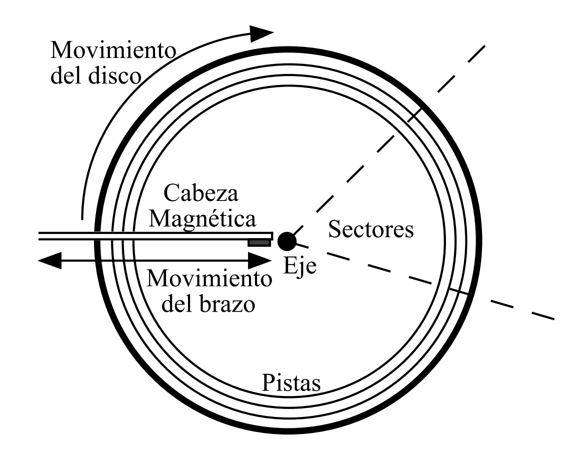


Figura 4.46. Esquema de funcionamiento de un disco duro

caras de cada uno de los platos está dividida en varias pistas (circunferencias concéntricas).

Al conjunto de pistas paralelas en diversas caras se le llama cilindro. Cada pista se divide en sectores (bloques de bytes), que son la unidad de lectura/escritura (véase la Figura 4.46). Cada sector tiene 512 bytes.

Tanto la lectura como la escritura en el disco se realizan mediante una cabeza magnética, que se desplaza de forma mecánica en sentido radial para situarse sobre la pista correspondiente. Las cabezas de lectura/escritura pueden colocarse en la superficie superior del plato, en la inferior o en ambas.

Los discos duros son dispositivos muy delicados y sensibles al polvo. Se presentan herméticamente cerrados en una caja con la circuitería y conectores adecuados (véase la Figura 4.47).





Figura 4.47. A la izquierda, un disco duro abierto: vemos la parte mecánica (discos magnéticos y cabezales) y la parte electrónica que la controla (circuitería). A la derecha, el disco duro herméticamente cerrado y listo para su conexión al PC

Acceso a la información en el disco duro

Para acceder a un dato del disco se ha de proporcionar la posición del sector en el que se encuentra. El sector se identifica mediante tres números: cilindro, cabeza y sector dentro de la pista. Las cabezas están asociadas a los platos; de hecho, cada una se mueve transversalmente al plato correspondiente y es capaz de posicionarse sobre él. El proceso es el siguiente:

- Las cabezas se mueven hasta que están encima del cilindro en cuestión.
- Después se activa la cabeza correspondiente.
- Se espera a que el sector pase por debajo de la cabeza.
- Se lee o escribe la información.

Como podemos apreciar, los discos magnéticos permiten el acceso directo a la información, pero el tiempo requerido para ello *no es constante*.

Características de un disco duro

A continuación, vamos a describir las características más importantes de un disco duro y que, de alguna manera, determinan la eficiencia de un disco magnético:

Capacidad: cantidad de bytes que podemos almacenar. Se mide en GigaBytes (GB). Actualmente, lo normal (en sobremesa o portátil) es tener instalado un disco entre 200 y 500 GB.

Tiempo de acceso: tiempo que tarda la cabeza en ponerse encima del sector donde se ubican los datos que debe leer o escribir. Tiene dos componentes: el *tiempo de búsqueda*, que es el tiempo que tarda en llegar a la pista, y el *periodo de latencia*, que es el tiempo que tarda en llegar el sector debajo de la cabeza. El tiempo de acceso se mide en milisegundos.

Velocidad de rotación: se mide en rpm (revoluciones por minuto). Cuanto mayor es la velocidad de rotación, más elevada es la velocidad de lectura/escritura (menor tiempo de acceso), pero también se produce mayor ruido y más calor (se puede necesitar un ventilador adicional).

Tamaño de la caché del disco: los discos duros incorporan una memoria que almacena la información que más probablemente se va a leer. Ésta es la memoria que va incluida en la controladora interna del disco duro. Los datos que se leen y escriben del disco duro se almacenan aquí en primer lugar. Si un disco duro está bien organizado, la serie de datos que se van a necesitar a continuación de una lectura estará situada en una posición físicamente contigua a la de la última lectura. Por eso los discos duros almacenan en la caché los datos contiguos, para proporcionar un acceso más rápido sin tener que buscarlos. De ahí la conveniencia de desfragmentar el disco duro con cierta frecuencia (sobre todo si se usa un sistema de ficheros basado en FAT, como los de algunos Windows, que tiende a fragmentar el disco).

Bus de conexión: hay distintos estándares de buses a considerar para interconectar un disco duro con el ordenador. Los más utilizados son IDE/EIDE/ATA y SATA, este último estandarizado en el año 2004. Para discos de altas prestaciones, se emplea la interfaz SCSI. En cuanto a la conexión de discos externos, lo habitual es emplear el bus USB 2.0, aunque también podemos emplear el bus FireWire. La elección del bus no es baladí, ya que determina la velocidad de transferencia, que es la velocidad con la que es posible transferir datos entre el disco y la memoria principal a través del bus.

4.4.4. Medios ópticos

Los medios ópticos son dispositivos capaces de leer datos digitales almacenados en un disco compacto. Estos dispositivos utilizan una lente que emite un haz de luz láser de baja frecuencia para leer los datos almacenados mediante la interpretación de las refracciones provocadas sobre su propia emisión. Si la luz rebota en la superficie del disco compacto, el ordenador lo registra como un 1, y si la luz no es reflejada (porque hay una “muesca”), el ordenador lo registra como un 0.

La lente se mueve del centro al borde del disco compacto, mientras éste gira, por la acción del motor interno del lector, de modo que se pueden leer datos en toda la superficie del disco compacto. Este tipo de medios permiten el acceso directo a la información.

CD-ROM

Un CD-ROM (acrónimo del inglés *Compact Disc - Read Only Memory*) es una unidad de disco compacto con una capacidad de almacenamiento *estándar* de 700 MB[18]. La velocidad de un CD-ROM define la tasa de transferencia de datos. Se expresa con un número seguido del signo "x", es decir, 2x, 4x, 16x, 24x... Este signo indica que se alcanza una velocidad de transferencia de 2, 4... veces la velocidad de un lector de discos compactos de audio, esto es 150 KBps. Por ejemplo, la tasa de transferencia de datos de un CD-ROM 24x será de 3.600 KBps., es decir, de unos 3,5 MBps. aproximadamente.

Los CD-ROM tienen pistas concéntricas en su superficie, similares a las que presenta un disco de vinilo, aunque mucho más finas y juntas. Solamente se utiliza una cara del CD-ROM para almacenar los datos. Los CD-ROM son soportes de sólo lectura y *no* son iguales que los CD-Audio, ya que éstos son otra variedad de discos compactos. Sin embargo, existen discos compactos que se pueden grabar:

- Los *CD-R* (*Compact Disc - Recordable*) son compatibles con los CD-ROM y funcionan en cualquier lector. Se puede escribir información en ellos *una sola vez*.
- Los *CD-RW* (*Compact Disc - ReWritable*) son compatibles con los CD-ROM, aunque algunos lectores no los leen bien. Se puede escribir (grabar) información en ellos varias veces. En la práctica, hay un límite al número máximo de veces que se puede reescribir datos en ellos, pero éste es muy grande y es muy probable que antes tengamos que tirar el CD debido al deterioro por su uso (rayas) que debido a que se haya alcanzado el límite de grabaciones.

DVD

El DVD (del inglés *Digital Versatile Disc*, aunque algunos lo llaman también *Digital Video Disc* porque el uso en películas es el más popular) es un disco óptico con el tamaño y apariencia física de un CD-ROM, pero con mayor capacidad de almacenamiento y velocidad de transferencia de datos. Un disco DVD se compone de dos discos pegados con un espesor de 0,6 mm. cada uno. Los DVD de doble capa utilizan un material semi-reflectante que permite separar las dos capas de información en cada cara del disco y hace que se puedan leer ambas capas desde una misma cara del disco. El lector láser accede a cada capa enfocando a diferente profundidad.

Si en los CD-ROM los múltiplos numéricos que indican la velocidad tomaban como base 150 KBps., en el caso de los DVD la base son 1.350 KBps., lo que significa que una unidad lectora de 16x permite una transferencia de datos de 21.600 KBps.

[18]Los hay de más capacidad, pero a costa de arriesgarse a que no se pueda leer. Al no ser estándar, no todos los lectores son capaces de leerlos.

(aproximadamente 21 MBps.). Como las velocidades de las unidades de CD se dan en múltiplos de 150 KBps., cada múltiplo de velocidad en DVD equivale a nueve múltiplos de velocidad en CD. En términos de rotación física (revoluciones por minuto), un múltiplo de velocidad en DVD equivale a tres múltiplos de velocidad en CD, así que la cantidad de datos leída durante una rotación es tres veces mayor para el DVD que para el CD, y la unidad de DVD 8x tiene la misma velocidad rotacional que la unidad de CD 24x.

En cuanto a la capacidad de almacenamiento, la diferencia con el CD reside en que el tamaño mínimo de una marca (un bit) en un DVD de una cara es de 0,44 micras, frente a las 0,83 micras del CD. Además, la distancia entre marcas en el DVD es de 0,74 micras, mientras que en un CD son 1,6 micras. Así pues, en el DVD se pueden hacer hasta algo más de 4 veces más marcas que en un CD, de ahí el incremento de capacidad. Obviamente, se necesita un láser más preciso, es decir, con menor longitud de onda, para leer/escribir datos en un DVD, de ahí que los lectores/grabadores de DVD puedan leer/escribir CD-ROM, pero a la inversa no es posible. Además, en un DVD se pueden utilizar las dos caras del disco y en cada cara puede haber dos niveles (capas).

Por lo tanto, los DVD se pueden considerar unidades de alta capacidad de sólo lectura, ya que pueden almacenar hasta 17 GB si usamos las 2 caras del disco con sus 2 respectivas capas. Existen discos DVD de distinta capacidad: 4,7 GB si usamos una cara del disco y una capa de datos; 8,5 GB empleando una cara y dos capas; 9,4 las dos caras con una sola capa y los de 17 GB que ya hemos mencionado.

Al igual que ocurre con los CD-ROM, existen formatos de DVD que permiten su escritura. Sin embargo, hay que tener cuidado, ya que no se ha impuesto un estándar, por lo que todavía existen formatos DVD incompatibles entre sí. En particular, los DVD+ son incompatibles con los DVD-: hay lectores/grabadores que sólo son capaces de leer/grabar uno de esos formatos. En la actualidad, lo normal es utilizar lectores/grabadores duales, capaces de trabajar con ambos formatos. Los DVD se pueden clasificar:

- Según su contenido:
  - DVD-Vídeo: Películas (vídeo y audio).
  - DVD-Audio: Audio de alta fidelidad.
  - DVD-Data: Datos cualquiera.
- Según su capacidad de regrabado:
  - DVD-ROM: Sólo lectura, manufacturado con prensa.
  - DVD-R: Grabable una sola vez.
  - DVD-RW: Regrabable.
  - DVD-RAM: Regrabable de acceso aleatorio.
  - DVD+R: Grabable una sola vez.
  - DVD+RW: Regrabable.
  - DVD-R DL: Grabable una sola vez de doble capa.
  - DVD+R DL: Grabable una sola vez de doble capa.

- DVD-RW DL: Regrabable de doble capa.
- DVD+RW DL: Regrabable de doble capa.

Más sobre el DVD

La especificación del formato de grabación del DVD-Vídeo es un subconjunto de la del DVD-ROM, lo que permite a los lectores de DVD-ROM reproducir discos DVD-Vídeo.

Para ello, necesitan descomprimir muy velozmente el formato MPEG-2 de representación binaria comprimida de vídeo y audio en el que están las imágenes.

La compresión del formato MPEG-2 se basa esencialmente en eliminar información redundante (zonas idénticas en fotogramas consecutivos y fotogramas replicados) e innecesaria (no perceptible por el ojo humano).

Las películas comerciales están "protegidas", esto es, *cifradas* con un algoritmo propietario. Sin embargo, al poco tiempo, este algoritmo fue "hackeado", lo que permite reproducir y copiar este tipo de DVD sin restricciones. Para ello, basta con instalar el *software* (DeCSS) que permite descifrar la información antes de descomprimir el MPEG-2 para su reproducción.

Es importante hacer notar que la legislación de algunos países (como España) impide la utilización de este tipo de *software*, que permite "saltarse" las protecciones establecidas por el propietario de los derechos de autor.

4.4.5. Otros periféricos de almacenamiento

Además de los discos duros y de los medios ópticos existen otros periféricos que sirven como dispositivos de almacenamiento:

Unidad ZIP. Una unidad ZIP es una unidad de almacenamiento removible de media capacidad y de bajo coste. Las primeras versiones alcanzaban los 100 MB, pero versiones posteriores ampliaron su capacidad hasta los 250 y 750 MB. Hasta hace unos años, se vendían cartuchos para almacenar 250 ó 750 MB, según modelos, y con interfaces USB, FireWire e IDE (ATAPI).

Estas unidades se convirtieron en los sucesores de los discos flexibles, sin embargo, la evolución y la caída de precios de los medios ópticos (grabadoras y consumibles CD-R, CD-RW y DVD) y, sobre todo, de los *pendrives* (memorias *flash* USB) y las tarjetas *flash* acabaron por eliminarlos del mercado y del uso cotidiano.

Discos externos. Aunque ya hemos mencionado algo sobre ellos en la sección 4.4.3, ya que mucha gente los considera como una variedad de discos duros, nos estamos refiriendo a los discos externos, susceptibles de ser conectados a puertos USB 2.0 o FireWire del PC, y de gran capacidad. Este tipo de discos son muy útiles para copias de seguridad y para transportar fácilmente de un sitio a otro

grandes volúmenes de datos. Los podemos encontrar desde los 30-40 GB (más pequeños y baratos) hasta los 2 TB (TeraBytes[19], más voluminosos y caros) pasando por los 160, 250, 320 ó 500 GB.

Compact Flash. Bajo esta denominación, estamos haciendo referencia no sólo a las memorias *CompactFlash* (CF) propiamente dichas, sino también a las memorias *Flash* USB (con una tecnología parecida *grosso modo*).

Las tarjetas CF son tarjetas de memoria que pueden ser conectadas directamente en una ranura *PC Card* con un adaptador (USB, por ejemplo). Compact Flash define una interfaz física, más pequeña que la interfaz PCMCIA-ATA, pero que es idéntica eléctricamente. Es decir, aparece para el dispositivo que la alberga como si fuera un disco duro de un tamaño definido y tiene un diminuto controlador IDE integrado en el mismo dispositivo CF. Actualmente, existen tarjetas desde los 8 MB hasta los 12 GB.

Por otra parte, los *pendrives* o memorias *Flash* USB son pequeños dispositivos de almacenamiento que utilizan memoria flash para guardar la información sin necesidad de baterías (pilas). Estas memorias son resistentes a los rasguños y al polvo que han afectado a las formas previas de almacenamiento portátil, como los CD y los disquetes. Desde el punto de vista electrónico, constan de un conector USB macho, que proporciona la interfaz física para la conexión al PC, un controlador USB de almacenamiento masivo, que provee la interfaz USB orientada a dispositivos de bloques, un circuito de memoria Flash NAND, que es el almacén de datos (este tipo de circuitos también se emplea en cámaras digitales), y un oscilador de cristal, que controla la salida de datos. Los podemos encontrar desde 256 MB hasta los 8, 16 e incluso 32 GB de capacidad.

4.4.6. Teclado y ratón

El teclado es el principal periférico de entrada del ordenador. Tiene su origen en los teletipos y en las máquinas de escribir eléctricas. Se suelen clasificar principalmente por la distribución de teclado de su zona alfanumérica, pues, salvo casos muy especiales, es común a todos los dispositivos y fabricantes (incluso para teclados árabes y japoneses).

Como todos los periféricos, los teclados también han evolucionado. De los primeros teclados de 83 teclas (utilizados en el PC XT –con procesador 8086 u 8088), se pasó a los teclados expandidos de 101/102 teclas y, posteriormente, al teclado Windows de 105 teclas. En la actualidad existen teclados multimedia que añaden más funcionalidad mediante teclas especiales que llaman a algunos programas en el computador, a modo de acceso directo, como pueden ser el programa de correo electrónico, la calculadora, el reproductor multimedia, etc.

También existen teclados ergonómicos, diseñados para dar una mayor comodidad para el usuario, ayudándole a tener una posición más relajada de los brazos, y teclados inalámbricos, más cómodos y comunicados a través de rayos infrarrojos, ondas de

[19] Un TeraByte equivale a 1.024 GB.



Figura 4.48. Arriba a la izquierda, una unidad ZIP y sus cartuchos. A la derecha, un *pendrive* USB. Abajo, un disco externo USB 2.0

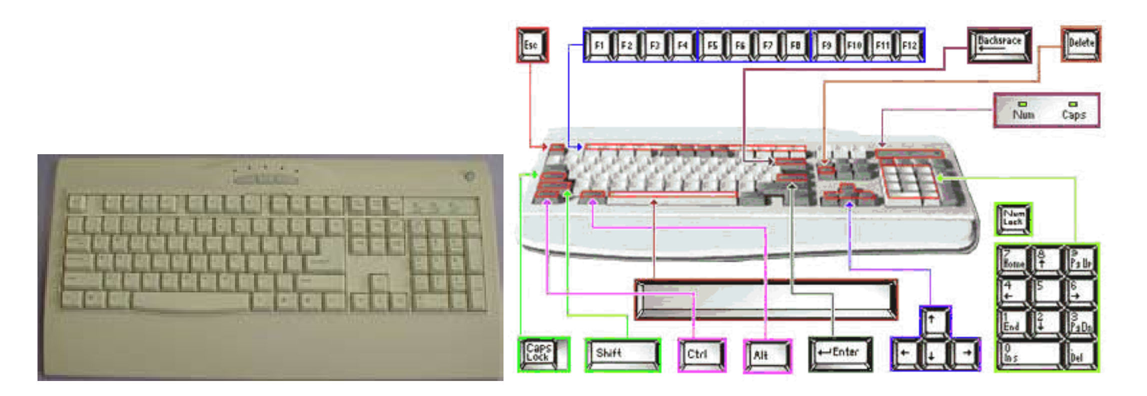


Figura 4.49. A la izquierda, un teclado convencional de 105 teclas. A la derecha, disposición de las teclas agrupadas por funciones

radio o mediante bluetooth (requieren pilas para su funcionamiento). Ten en cuenta que existen modelos alternativos de teclado (el más usado es el modelo QWERTY, llamado así por la disposición de estas teclas en la fila superior de letras comenzando por la izquierda) y que los teclados de otros países, aun siendo el mismo modelo, no tienen la misma disposición de teclas ("ñ", acentos, etc.).

El ratón es un periférico de entrada que resulta imprescindible para el trabajo con entornos gráficos de usuario (se utiliza para mover un cursor por la pantalla, apuntar

y seleccionar objetos…) tan comunes hoy en día (en los ordenadores personales de Apple con OSX, en los PC con Windows y Linux –KDE, por ejemplo–, etc.). En la actualidad existen diferentes modelos de ratón en función del número de botones (ruedas) del mecanismo o de la forma de conexión (PS2, USB).

Los ratones más habituales son los de tres botones, en los cuales el botón central es una rueda que permite desplazarse por el contenido de las ventanas (a este tipo de acción se le llama *scroll*).

En cuanto al mecanismo, existen ratones mecánicos (utilizan el movimiento de una bola), ópticos (disponen de un láser) o *trackballs*. Al igual que ocurre con los teclados, existen ratones que pueden conectarse de forma inalámbrica mediante infrarrojos, ondas de radio o *bluetooth* (y que requieren, igualmente, de pilas para su funcionamiento).

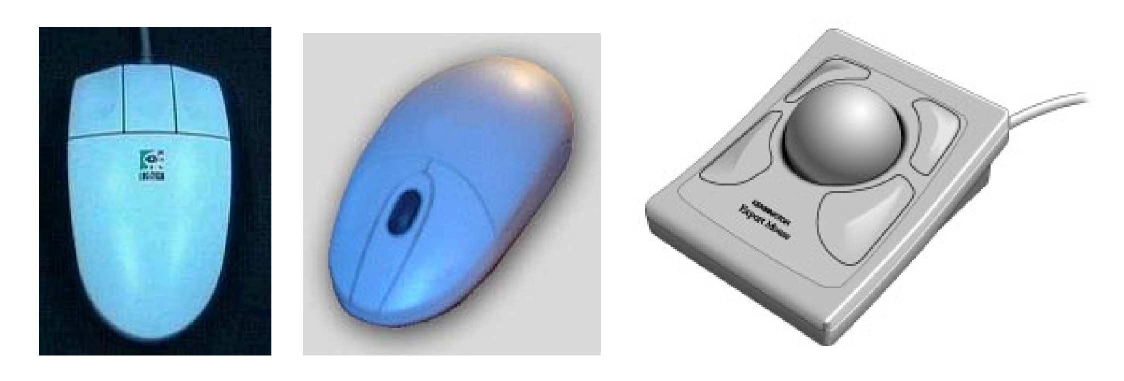


Figura 4.50. A la izquierda, un modelo clásico de ratón con 3 botones. En el centro, un ratón donde el tercer botón (central) es una rueda (para *scroll*). A la derecha, un *trackball*

4.4.7. Impresora

La impresora es un periférico de salida, que permite obtener una copia impresa (textos y gráficos) de la información almacenada en el ordenador. En este sentido, actúa como una máquina de escribir, pero es mucho más, ya que permite la impresión de gráficos. La combinación del uso de un programa procesador de textos junto con la impresora ha permitido "jubilar" la máquina de escribir.

Las impresoras se pueden conectar al ordenador mediante el puerto serie, paralelo o USB. Actualmente, el puerto USB es el más utilizado. También es posible conectar las impresoras en red (por ejemplo, en una red tipo Ethernet), pudiendo, de esta forma, servir a un conjunto de usuarios en una red local. Existen diferentes tipos de impresoras:

- Impacto, normalmente tipo matriz de puntos o agujas. Se basan en la fuerza de impacto para transferir tinta al medio (papel), de forma similar a las máquinas de escribir. Se encuentran limitadas a reproducir texto (y también líneas) y sólo se utilizan en entornos donde se trabaja con papel autocopiable (como papel de calco, para obtener más de una copia simultáneamente).



Figura 4.51. Arriba: a la izquierda, una impresora de inyección de tinta; a la derecha, esquema de funcionamiento del mecanismo de inyección de tinta. Abajo, una impresora láser

También sirven para escribir en sobres especiales que contengan información confidencial (clave secreta de un banco, por ejemplo). Son impresoras antiguas, su calidad no es muy buena y suelen ser bastante ruidosas.

- Inyección de tinta. Utilizan inyectores que producen burbujas de tinta muy pequeñas que posteriormente se convierten en pequeñísimas gotitas que se lanzan contra el papel. Los puntos formados son del tamaño de los pequeños píxeles. Las impresoras de inyección pueden imprimir textos y gráficos de alta calidad de manera casi silenciosa. Permiten utilizar tinta de colores y, en general, son silenciosas, rápidas y de buena calidad.

  Las impresoras de inyección de tinta son extremadamente baratas. Sin embargo, los consumibles, esto es, los cartuchos de tinta, son extremadamente caros. Una hoja impresa en blanco y negro puede costar entre 3 y 6 céntimos, mientras que una a color entre 10 céntimos y algo más de un euro. Las impresoras láser son más caras que las de inyección de tinta, pero los consumibles son más baratos.

- Láser. Trabajan utilizando el principio *Xerographic* que consiste en adherir tóner

a un tambor de impresión sensible a la luz (mecanismo similar al de una fotocopiadora). En este caso, se trata de la luz generada por un láser. Mediante la electricidad estática se transfiere el tóner al medio de impresión (normalmente papel) al cual se une gracias al calor y la presión.

La impresora es más cara que la de inyección de tinta: el doble, o más, los modelos en blanco y negro; los buenos modelos de gama media-alta que imprimen en color tienen precios hoy en día rondando los 200-600 €. Sin embargo, el consumible (tóner) es más barato: una página impresa puede salir por entre 1 y 5 céntimos. La mayoría son capaces de imprimir directamente ficheros en formato *Postscript*. Son silenciosas, muy rápidas y obtienen resultados de una gran calidad.

4.4.8. Escáner

El escáner es un dispositivo de entrada de datos al ordenador que permite, mediante el uso de la luz, digitalizar imágenes y textos.



Figura 4.52. Un escáner

La imagen o el papel con el texto a digitalizar se coloca en la superficie de cristal del escáner, enfrentada al bloque lector y al cabezal lector compuesto por el CCD, el sistema de iluminación y un conjunto de lentes que se desplazan explorando la imagen. La luz reflejada se convierte en energía eléctrica por los sensores, y la velocidad del movimiento del cabezal lector es la que proporciona la resolución. Cuanto menor sea la velocidad del lector, más información se extraerá de la imagen digitalizada.

El programa de control del escáner (el fabricante suele proporcionar uno, pero en Internet hay varios a nuestra disposición; algunos de ellos libres) permite visualizar la imagen digitalizada y guardarla en algún formato típico (JPEG, PNG, TIFF, etc., véase el capítulo 3, "Codificación de la información"). En este sentido, es como hacer una *fotocopia digital* de la imagen o texto escaneado.

Cuando escaneamos textos, en la mayoría de las ocasiones no queremos una mera imagen, sino que nos gustaría que el ordenador pudiese "leer" o, mejor dicho, "reconocer" los caracteres que conforman el texto para poder guardarlo en algún formato estándar para procesadores de texto (como PDF o el formato Open Document de OpenOffice). Ahora bien, el escáner por sí solo no es capaz de reconocer texto. Si queremos hacerlo, se necesita un programa de OCR (*Optical Character Recognition*) que sea capaz de identificar los caracteres que componen el texto digitalizado[20]. En Internet podemos descargarnos programas libres de OCR.

[20]Los programas de OCR tienen elevados porcentajes de acierto si se trata de letra impresa. Con la escritura manuscrita su desempeño no es tan bueno: hoy en día es un campo de investigación.

Actualmente, la conexión del escáner con el ordenador se suele realizar, mayoritariamente, mediante el puerto USB, aunque también es posible realizarla mediante SCSI. Existen diferentes tipos de escáneres:

Rodillo: idéntico al funcionamiento del escáner de un fax.

De mano: actualmente están obsoletos. En su momento eran muy económicos, pero de muy baja calidad.

Planos: son similares al mecanismo de las fotocopiadoras. Actualmente son los más extendidos.

Otros tipos: existen tipos de escáneres especializados en documentos específicos (tales como diapositivas, textos de un libro completo, negativos, etc.).

4.4.9. Dispositivos multifunción

Estos dispositivos son periféricos de entrada/salida que se conectan a la computadora y que son capaces de desempeñar varias funciones. Las principales funciones que pueden estar disponibles en estos dispositivos son:

- impresora,
- escáner,
- fotocopiadora, ampliando o reduciendo el original,
- lector de tarjetas (por ejemplo, *Compact Flash*, o alguna de las variedades empleadas en cámaras fotográficas digitales),
- fax.

Estos dispositivos multifunción o MFP (siglas del inglés *Multi Function Printer/Product/Peripheral*) pueden funcionar como un periférico más del ordenador o bien de forma autónoma. Por ejemplo, para fotocopiar un documento no hace falta encender el ordenador. Normalmente, se conectan al PC a través de un puerto USB.

4.5. ELEGIR UN ORDENADOR

En la actualidad, hay una serie de componentes estándar que están presentes en prácticamente todos los ordenadores personales, ya sean de sobremesa, ya sean portátiles, tales como procesador, placa base, caja, disco duro (obviamente, ya que son los componentes fundamentales), tarjeta de sonido, tarjeta de comunicaciones (Ethernet); buses, tales como PCI o USB; y periféricos, tales como teclado, ratón e impresora.

No obstante, las prestaciones que ofrecen los diversos modelos que podemos encontrar en el mercado de estos productos varían considerablemente y, por ende, su precio. La primera cuestión que tenemos que tener presente a la hora de comprar un PC es pensar cuál va a ser su finalidad: ¿va a ser para ocio, para el trabajo, para dar servicios (servidor)? ¿qué clase de trabajos vamos a realizar con él: ofimática, contabilidad, programación, diseño 3D, CAD. . . ?

Por otro lado, hay otros componentes que no forman parte de lo que podríamos llamar una "configuración estándar", bien por su elevado precio, bien por su especificidad, y que pueden ser necesarios según la funcionalidad que le demos al ordenador que vamos a comprar. Por ejemplo, si queremos montar un servidor web para acceder a un elevado número de documentos almacenados en disco, probablemente optaremos por comprar un disco SCSI (junto con la adquisición de la controladora correspondiente), que garantiza altísimas tasas de transferencia de datos. Por tanto, en función del uso que se le vaya a dar, elegiremos los componentes del PC. Otro ejemplo: no necesitará la misma tarjeta de sonido (o procesador digital de la señal) un programador que un músico o un ingeniero de sonido que se dedican profesionalmente a la edición y producción de audio.

Si bien hasta hace unos pocos años la mayoría de ordenadores personales de sobremesa que se vendían eran los populares "clónicos" (selección a la carta de los componentes del PC) que ensamblaban los técnicos de las tiendas de informática, en la actualidad se venden igualmente muchos equipos "de marca": ordenadores personales con una configuración y componentes elegidos de serie y con el sello oficial de alguna empresa o multinacional informática, tal como IBM, Hewlett-Packard o Dell. En lo que se refiere a portátiles, el mercado ha estado copado mayoritariamente por los ordenadores "de marca". Esto se entiende dada la especificidad que tradicionalmente ha tenido este tipo de ordenadores (algo en cuestión en la actualidad) y sus peculiaridades técnicas (circuitería específica, batería, tipos de placa, etc.).

¿Por qué antes se vendían mayoritariamente los PC de sobremesa "clónicos" y ahora no tanto? Fundamentalmente por una razón de precio. En su momento, el ensamblaje de componentes a la carta abarataba los costes del equipo: si se hacía correctamente[21] el usuario disfrutaba de un PC con unas prestaciones que nada tenían que envidiar a los "de marca" a un precio mucho más asequible. El riesgo que se corría era un mal funcionamiento del equipo por no realizar los controles de calidad correspondientes (que sí pasan los "de marca").

En la actualidad, la competencia ha ido eliminando a los malos profesionales y ocurre que las tiendas que venden "clónicos" se esmeran más en sus controles y eligen componentes de mayor calidad. Pero también ocurre que han bajado los precios de los componentes y que las empresas de informática han alcanzado mejores acuerdos para el ensamblaje de sus productos. Todo ello ha provocado que los precios de los ordenadores "de marca" y "clónicos" se hayan acercado bastante. Si a eso le unimos las amplias garantías y el servicio técnico eficaz que ofrecen la mayoría de empresas y multinacionales de informática, nos encontramos con que las ventas de ambos tipos se han aproximado bastante.

En resumen, vale la pena prestarle atención a la tarea de elegir un ordenador. Tanto sea un ordenador "de marca" como un "clónico", deberemos prestar atención, fundamentalmente a:

- Garantía de la marca o de la tienda (referencias, buena reputación...).
- Garantía del equipo y componentes (tiempo y condiciones), servicio técnico.

[21] Y esto no siempre ha sido así.

- Procesador: hay modelos más caros y potentes que otros. Hemos de ver si realmente necesitamos esa potencia o podemos ahorrarnos un dinerillo.
- Monitor (marca, garantía, tamaño y resoluciones).
- Tipo de componentes: tarjeta gráfica (ha de ser buena y con suficiente memoria propia si vamos a usar aplicaciones 3D, si no, podemos ahorrar dinero comprando una más modesta y con memoria compartida), tarjeta de sonido (sólo profesionales) y disco duro (interfaz, capacidad, velocidad de rotación y tiempo de acceso).
- Placa base: buses y conectividad. Seguro que tendremos buses PCI y USB 2.0. Para USB hay que mirar el número de puertos (para conexiones) y su posición (si son frontales en un sobremesa, mejor). Pueden ser necesarios, además, los buses PCI Express (si empleamos una gráfica de última generación) y FireWire (sigue siendo mayoritariamente usado para la conexión de cámaras digitales de vídeo, por ejemplo). El bus SCSI podemos incorporarlo mediante una tarjeta de expansión si fuese necesario (¡ojo! es caro y también lo son los dispositivos SCSI).
- En cuanto a la interfaz de conexión para el disco duro, hemos de tener presente que los discos IDE/EIDE/ATA son los más baratos, pero también los más lentos; aunque sus prestaciones entre 100 y 130 MBps. resultan más que suficientes en muchos casos (ofimática, navegación por Internet, programación, etc.). Hoy en día, los discos SATA constituyen una alternativa real, ya que ofrecen mejores prestaciones (en torno a 150 MBps.) y su precio es sólo ligeramente más alto. Los discos SATA-2 han bajado algo su precio (aunque aún son más caros) y son el doble de rápidos que los SATA.

4.5.1. Para la oficina

Si estamos pensando en adquirir un ordenador para realizar tareas de oficina, como podría ser el PC para el puesto de administrativo en una empresa u organismo, debemos considerar que las aplicaciones que principalmente se van a ejecutar son: procesador de textos, hoja de cálculo, navegador web, programas de contabilidad o facturación y, tal vez, aplicaciones propias de gestión que actúen contra una base de datos corporativa. En este sentido, nuestras recomendaciones son:

- Podemos ahorrar dinero con el procesador, ya que no necesitamos el más potente del mercado. Probablemente un procesador de gama media o media/baja sea suficiente; algo similar a un modelo Celeron D (Intel) o Sempron (AMD), tal vez.
- También podemos ahorrar dinero con la tarjeta gráfica, ya que, aunque en la actualidad prácticamente todas incluyen aceleración 3D, hay varios modelos de bajo coste con poca memoria propia (usan memoria compartida, o sea, la RAM).
- Teniendo en cuenta que las aplicaciones de ofimática y los navegadores consumen más memoria RAM que ciclos de CPU, se recomienda no cicatear la

memoria RAM, sino que cuanta más, mejor. Un mínimo entre 1 y 2 GB estaría bien.

- En cuanto a disco duro, se sugiere adquirir un disco de gran capacidad (mínimo 250 GB) y, a ser posible, SATA. También puede ser interesante usar un disco externo USB 2.0 para efectuar copias de seguridad y transportar grandes volúmenes de información fácilmente de un sitio a otro.
- Un buen monitor de 17" TFT es recomendable, ya que así conseguimos mejores resoluciones de trabajo y evitamos la fatiga visual del trabajador en su puesto.
- En un entorno de oficina, donde se realizan varias copias en papel a lo largo del día, se recomienda la adquisición de una impresora láser debido al menor coste de los consumibles (la diferencia de precio con otros tipos de impresora se amortizará rápidamente). Quizás, podría plantearse la adquisición de una impresora de impacto (por lo del papel autocopiable en facturas).
- Podría ser interesante adquirir un escáner (que pueda conectarse a USB) y su empleo con *software* de OCR, para poder pasar fácilmente a formato electrónico editable (procesador de textos) documentos impresos en soporte papel.
- Como alternativa para las copias de seguridad, podemos pensar en usar cintas magnéticas o DVD regrabables (DVD-RW).
- Normalmente, convendrá que la placa lleve integrada una tarjeta Ethernet (tal vez WiFi) para su conexión a la red local de la empresa.

4.5.2. Para aplicaciones lúdicas

En este caso, estamos pensando en adquirir un ordenador para nuestro ocio, concretamente para la ejecución de juegos para PC. Es como si hubiésemos decidido comprarnos "algo más" que una consola. Obviamente, si tan sólo se trata de jugar, los autores recomendamos la compra de un artilugio específico para ello: una videoconsola. Pero, claro, por un poco más de lo que cuestan las consolas de última generación tenemos un buen ordenador personal que sirve igualmente para jugar con juegos de última generación y nos da más posibilidades. Nuestras recomendaciones en esta situación son:

- Habría que considerar la compra de un procesador de gama alta, algo en la línea de un Dual o Quad Core, o tal vez Pentium 4 de última generación, ya que los juegos requieren microprocesadores rápidos. Para los profesionales de los juegos tanto Intel como AMD ofrecen sus "Extreme Edition".
- Por supuesto, habría que adquirir una tarjeta gráfica avanzada, con aceleración 3D y mucha memoria *propia* (no compartida); en torno a los 256 ó 512 MB. Una buena tarjeta 3D puede permitir comprar un microprocesador de menor potencia. También hay que tener presente el tipo de API 3D que usan los juegos que vamos a ejecutar para ver si la tarjeta le da soporte (por ejemplo, no todas las tarjetas del mercado dan soporte a la versión 10.0 del API DirectX).

- Muchos juegos requieren de elevadas resoluciones para el óptimo disfrute de sus gráficos y texturas. Es por ello que se recomienda un monitor de 19 ó 21” TFT.
- Un disco duro “estándar” de 160 ó 200 GB será suficiente en la mayoría de las ocasiones.
- Si se ha de comprar una impresora, se recomienda adquirir una de inyección de tinta, ya que son más baratas. No obstante, conviene hacer una previsión acerca de cuánto vamos a imprimir, ya que en función del precio de los consumibles a lo mejor podría resultar conveniente comprar una láser.
- Conviene comprar una buena tarjeta de sonido. Hay que fijarse en las características de los juegos para seleccionar la más conveniente.
- Como la mayoría de los juegos vienen en CD o DVD, se recomienda comprar una regrabadora dual (soporta ambos formatos de grabación) de DVD: nos sirve como unidad lectora y también para copias de seguridad en DVD-RW.
- Algunos juegos incluyen la posibilidad de jugar con otros usuarios en red, por lo que se recomienda contratar conexión a Internet de banda ancha (módem ADSL o cable-módem). Según la modalidad contratada se necesitará tarjeta Ethernet o no; aunque los módems usados en ambas modalidades se pueden conectar habitualmente al puerto USB.

4.5.3. Para gráficos y multimedia

Si vamos a trabajar principalmente con aplicaciones multimedia (imagen y sonido) o aplicaciones 3D (CAD, modelado sólido, animación. . .), conviene que tengamos en cuenta lo siguiente:

- No deberíamos escatimar gastos con el procesador. Optaremos por un procesador de gama alta con especial recomendación por uno de doble núcleo, o, incluso, de cuatro núcleos de muy reciente aparición. Las aplicaciones de edición de vídeo, sonido e imagen requieren uso intensivo de CPU para realizar ciertas transformaciones.
- Obviamente, compraremos una buena tarjeta gráfica con aceleración 3D y bastante memoria *propia* (256 MB mínimo). Nuevamente, hay que fijarse en los requisitos de los programas 3D con los que vamos a trabajar (por si fuese necesario que la tarjeta tuviese que ser compatible con DirectX 10.0, por ejemplo).
- Como en el caso anterior, se recomienda un buen monitor de 19 ó 21” TFT.
- La edición de vídeo requiere, normalmente, del uso de ingentes cantidades de espacio en disco duro. También es importante la velocidad de transferencia de datos. Por tanto, recomendamos adquirir un disco con gran capacidad (400 ó 500 GB) y, a ser posible, SATA o SATA-2 (por tanto, la placa base debiera soportar estas interfaces).
- Para procesar el sonido desde un punto de vista profesional, es necesario adquirir una tarjeta de sonido de gama alta, con soporte de Dolby y DTS, con una buena

resolución de sonido, alta fidelidad y filtro de ruidos. Es importante que tenga conexiones (puertos) Digital S/PDIF Output/Input, conector para cable coaxial o fibra óptica y MIDI (por ejemplo, para la conexión de periféricos específicos tales como un teclado MIDI, con el que se puede componer en formato digital).

- Para copias de seguridad se puede emplear DVD regrabables o bien un disco duro externo USB 2.0.
- Si se trabaja con programas de modelado sólido, CAD o tratamiento de imágenes, es muy probable que necesitemos realizar impresiones con calidad fotográfica y alta resolución. Se recomienda la compra de una impresora láser color de gama alta (se supone que vamos a hacer muchas impresiones y la amortizaremos pronto).En el caso de que deseemos imprimir pósteres o carteles, es probable que se requiera la adquisición de un dispositivo trazador o *plotter* (por ejemplo, si se trata de una agencia de publicidad).
- Es recomendable que la placa base del ordenador que compremos tenga bus FireWire y tengamos el correspondiente puerto de conexión (cámara digital de vídeo).
- En el mundo del diseño gráfico y la edición multimedia, podemos considerar la compra de un Apple Macintosh en lugar de un PC.

4.5.4. Para un estudiante de informática

Si estás pensando en comprar un PC que te sirva de apoyo durante tus estudios de Ingeniería (Técnica) Informática, es decir, para realizar prácticas, trabajos y estudiar, en esta sección te aconsejaremos qué es lo más conveniente con el fin de intentar ahorrar el máximo dinero posible. Ten en cuenta que nosotros nos centramos exclusivamente en los estudios. Si, además, quieres usar el ordenador para ocio, juegos o edición multimedia, serían aplicables muchos de los consejos expuestos en las secciones anteriores.

- Puedes ahorrar dinero con el procesador, ya que basta con que compres uno de gama media-alta. Por ejemplo, un procesador Dual Core básico de Intel o similar de AMD sería suficiente (y, ahora, con la llegada de los procesadores de cuatro núcleos han bajado de precio).
- También puedes ahorrar dinero con el monitor. Te bastaría un buen monitor de 17" TFT para trabajar con buenas resoluciones. No obstante, y puesto que para programar (nos referimos a programas de cierta envergadura) se suele trabajar con varias ventanas de edición simultáneamente, puede ser conveniente comprar un monitor de 19" o 21" si el presupuesto te lo permite.
- En cuanto a disco duro, te recomendamos un "estándar" habitual y económico en los PC actuales: disco SATA de 160 GB. Por supuesto, si hay dinero puedes optar por comprar un disco de mayor capacidad y con interfaz SATA-2, por ejemplo, pero esto no sería estrictamente necesario.

- Para copias de seguridad, te recomendamos la utilización de DVD regrabables. Prácticamente todos los PC que se venden vienen equipados con regrabadora dual de DVD y, de esta manera, con 5 ó 6 DVD puedes respaldar entre 20 y 25 GB fácilmente. Si el volumen de datos que quieres respaldar es mucho mayor, entonces podría resultar conveniente optar por la compra de un disco externo USB 2.0 (o FireWire) de capacidad media (en torno a los 200 GB).

- Es conveniente que dispongas de algún dispositivo que permita trasladar cómodamente datos entre casa y la Universidad. Estamos hablando de un dispositivo que tenga conexión a USB (que es un bus que encuentras en todos los PC actuales, incluyendo los de las aulas informáticas) y que sea pequeño (como un llavero) y fácil de transportar. Te sugerimos la compra de un *pendrive* o memoria flash USB, de 1 ó 2 GB como mínimo.

- También es importante que dispongas de una buena conexión a Internet desde tu casa (banda ancha, pero no el máximo: una conexión de 1 ó 2 Mbps. de bajada y 320 Kbps. de subida podría estar bien). De esta manera podrás acceder a los servicios que proporciona la UJI, tales como la pasarela Webmail para gestionar tu buzón de correo electrónico o el Aula Virtual para acceder a las páginas de las diversas asignaturas y bajarte apuntes o subir prácticas. Puesto que hablamos de banda ancha, lo normal es que contrates cable o ADSL; por tanto tendrás que utilizar o bien un cable-módem o bien un módem ADSL. Ten en cuenta las características de ambos aparatos, ya que, si bien normalmente ambos permiten la conexión al PC mediante USB, en algunas ocasiones puede resultar conveniente conectarlos a través de una tarjeta Ethernet[22].

- Si deseas adquirir una impresora, algo que te recomendamos si quieres imprimir en tu casa los apuntes o enunciados de prácticas que muchos profesores dejamos en las páginas web de nuestras asignaturas, te recomendamos una impresora de inyección de tinta, ya que es más barata. No obstante, si crees que vas a utilizarla a menudo, puede resultar conveniente adquirir una impresora láser. Fíjate en el precio de los consumibles. Si tu presupuesto está ajustado y deseas comprar una láser (por los consumibles), podrías plantearte comprar una en "blanco y negro" (en realidad, escala de grises): son algo más baratas que las de color.

4.5.5. ¿"De marca" o "clónico"?

Al principio de esta sección ya hemos comentado algunas cosas al respecto de las diferencias entre ambos tipos de ordenadores personales. A continuación, vamos a resumir sus principales ventajas:

(a) Ventajas de los ordenadores "de marca":

  - Mayor control de calidad.
  - Instrucciones de uso e instalación. Manuales traducidos.

[22] Algunos modelos de módem permiten ambos tipos de conexiones, pero si los conectas mediante USB es posible que sistemas operativos alternativos a Windows, como Linux, no los reconozcan y, por tanto, no funcionen en ellos.

- Servicios especiales (instalación en casa, garantía extendida, servicio técnico, asistencia telefónica. . .).
- Sistema operativo preinstalado (Windows, habitualmente). Esto puede resultar una desventaja en algunas ocasiones, ya que pagamos la licencia correspondiente y, a lo mejor, queremos instalar solamente Linux (o ambos).

(b) Ventajas de los ordenadores "clónicos":

- Más baratos.
- Posibilidad de elegir componente a componente: "ordenador a la carta".
- Componentes estándar.
- Sistema operativo preinstalado o no, a voluntad.

En general, la ventaja mayor que ofrecen los "clónicos" sobre los "de marca" reside en su mayor flexibilidad, ya que puedo elegir sólo los componentes que deseo (incluso sus marcas), y en su menor precio (puedo buscar aquellas marcas de componentes que mejor relación calidad/precio ofrecen).

Sin embargo, hoy en día esto ya no es tan así. Tal como comentábamos al principio de esta sección, las tiendas de informática que venden "clónicos" han ido incrementando sus controles de calidad y añadiendo mejores componentes (también más caros) y las empresas y multinacionales ofrecen una amplia gama de productos para elegir y consiguen "ensamblar" ordenadores más baratos.

Por tanto, en la actualidad no resulta descabellado optar por un ordenador "de marca", ya que no resulta oneroso lo que pagamos de más. Algunos fabricantes, incluso, permiten solicitar un ordenador de sobremesa "a medida" a partir de determinados componentes (dentro de unos ciertos márgenes de elección) y te dicen cuál es su precio final resultante. Por ejemplo, éste es el caso de Dell, que permite elegir un ordenador "a medida", tanto sobremesa como portátil, desde su página web: `http://www.dell.es`.

También hay una pregunta que se nos formula de forma recurrente en clase por parte de los alumnos desde hace un par de años y a la que quisiéramos dar respuesta aquí: "quiero comprarme un ordenador y no sé si adquirir un sobremesa o un portátil, ¿tú qué opinas?". La respuesta a esta pregunta hay que plantearla desde tres puntos de vista: movilidad, autonomía[23] y precio/prestaciones.

Movilidad: Si lo que deseamos es tener un ordenador personal con el que se pueda trabajar en diversos ámbitos (casa, Universidad, etc.) y con el que podamos desplazarnos fácilmente de un sitio a otro, la elección resulta obvia. Además, con el auge actual de las redes inalámbricas (WiFi) y la cobertura de la que se dispone en los recintos universitarios, con un portátil podremos conectarnos a Internet desde casi cualquier lugar del campus (obviamente, los portátiles, habitualmente, vienen equipados con tarjetas inalámbricas).

[23] En su acepción de "tiempo máximo que puede funcionar un aparato sin repostar" (diccionario de la RAE).

Autonomía: Cuando hablamos de "autonomía" estamos haciendo referencia a la posibilidad de tener el ordenador funcionando durante varios días seguidos sin problemas. Esto es algo que, normalmente, sí que se puede realizar sin problemas con un ordenador de sobremesa, pero que presentaba mayores problemas con un portátil, debido sobre todo a problemas de calentamiento (y a algunos problemas con los discos duros). Sin embargo, en la actualidad esto ya no es así: un portátil puede estar funcionando de manera ininterrumpida (conectado a la corriente, por supuesto) durante muchas horas e incluso algunos días, sin mayores problemas.

Precio/prestaciones: Si bien es cierto que los ordenadores de sobremesa son más baratos que los portátiles de similares características, también lo es que las diferencias en cuanto a precio se han reducido considerablemente. La tecnología empleada en la fabricación de portátiles es mucho más barata que hace unos años y, hoy en día, no resulta descabellado para el consumidor medio adquirir un portátil bien equipado. En cuanto a prestaciones, tampoco se observan grandes diferencias. Actualmente, podemos encontrar para portátiles:

- procesadores de múltiples núcleos (última generación) con unos pocos meses de diferencia respecto a su aparición en el mercado para equipos de sobremesa,
- tarjetas gráficas con aceleración 3D de última generación y hasta 512 MB de memoria propia,
- discos duros de gran capacidad (desde 40 hasta 320 y 500 GB),
- regrabadoras de DVD de hasta 8x, y
- módulos de memoria SDRAM DDR2 a un precio equiparable al de los módulos para equipos de sobremesa, hasta el punto de que lo "normal" es tener portátiles equipados con entre 1 y 2 GB de RAM.

Por supuesto, la decisión final queda en manos del lector, pero, desde luego, ya no se observan, en general, grandes diferencias entre portátiles y equipos de sobremesa. Nuestra recomendación para un estudiante de informática es adquirir un portátil, siempre que se esté dispuesto a gastarse un poco más de dinero para obtener prestaciones similares. El estudiante podrá trabajar en casa y en el campus universitario, con conexión a Internet sin mayores problemas.

CAPÍTULO 5

REDES DE COMPUTADORES

BIBLIOGRAFÍA

Algunos de los datos que aparecen en el texto han sido consultados y verificados con información proveniente de la *Wikipedia*:

WIKIPEDIA, LA ENCICLOPEDIA LIBRE: `http://es.wikipedia.org`. Fechas de consulta: enero, febrero y marzo de 2008.

WIKIPEDIA, THE FREE ENCYCLOPEDIA: `http://en.wikipedia.org`. Fechas de consulta: enero, febrero y marzo de 2008.

5.1. ¿QUÉ ES LA TELEINFORMÁTICA?

Con el nombre de *teleinformática*, o también *telemática*, se conoce al conjunto de técnicas y métodos de procesamiento distribuido que permiten realizar un tratamiento automático de la información a distancia usando técnicas y recursos informáticos y de comunicaciones. Estas técnicas constituyen el núcleo básico de lo que llamamos *redes de computadores*.

En general, se considera que una red de computadores es cualquier conjunto de ordenadores que pueden comunicarse entre sí. Por lo tanto, una red puede abarcar únicamente dos ordenadores o varios millones.

Además, en muchos casos, distintas redes pueden estar comunicadas entre sí. Esto forma las denominadas "interredes" (traducción del inglés *internets*), la más famosa de las cuales es la propia red mundial de redes conocida simple y popularmente por el nombre de *Internet*.

En la actualidad las redes de ordenadores son ubicuas y forman parte incluso de nuestro tejido social. Las redes informáticas aparecieron hace unos cuarenta años y sus aplicaciones son innumerables, desde las tareas de control, banca, computación distribuida. . . hasta el ocio (juegos en red) y las relaciones humanas (correo electrónico, mensajería instantánea, foros, *chat*, etc.).

Entre las ventajas que proporcionan las redes de ordenadores, podemos destacar

las siguientes, desde un punto de vista empresarial:

- *Permiten compartir recursos.* Por ejemplo, el acceso a documentación, programas o máquinas situadas en lugares apartados geográficamente o la publicación de bases de datos.
- *Aumento de la fiabilidad*, ya que podemos replicar recursos cómodamente. Por ejemplo, si los ficheros se mantienen en varias máquinas, cuando una de ellas falla se pueden emplear otras.
- *Ahorro.* La relación precio/rendimiento es mucho mejor en los ordenadores pequeños que en los grandes. Así, es preferible que cada usuario tenga su propio ordenador a que tenga una terminal de un ordenador grande. Esto, además, ha contribuido a extender el modelo cliente/servidor, en el cual los recursos costosos se comparten.
- *Proporcionan escalabilidad.* Para obtener más potencia basta, en principio, con añadir nuevos ordenadores a la red.
- *Posibilitan el trabajo a distancia*, facilitando las comunicaciones. Las redes son un medio de comunicación que permite trabajar juntos a empleados separados geográficamente.

Si miramos a las redes de ordenadores desde un punto de vista personal, como individuos que realizan actividades y se relacionan con otros, las redes de ordenadores también proporcionan ventajas:

- *Permiten el acceso a información remota* de todo tipo, especialmente con el advenimiento y ulterior desarrollo del servicio de *World Wide Web* (WWW). Por ejemplo, podemos acceder a diccionarios (como el de la RAE: `http://www.rae.es/`) o enciclopedias (como la Wikipedia: `http://es.wikipedia.org/wiki/Portada`).
- *Permiten nuevas y diversas maneras de comunicación interpersonal.* Por ejemplo, el correo electrónico, la mensajería instantánea o la videoconferencia.
- *Permiten nuevas maneras de entretenimiento* y relación social. Por ejemplo, juegos en red (como *Unreal Tournament*), foros de personajes públicos (*blogs*) o redes sociales (como *Twitter* –`http://twitter.com/`– o *Facebook* –`http://es.facebook.com/`–).
- *Permiten la compartición de datos on-line* entre varios usuarios. El ejemplo típico son las conocidas *redes de pares* (P2P), como la del *E-mule* o *BitTorrent*, por ejemplo.

5.2. HITOS HISTÓRICOS

En los años sesenta (siglo pasado) surgieron los primeros sistemas que podemos considerar telemáticos: un ordenador central al que se conectan terminales "tontos". En aquella época el precio de los ordenadores era muy elevado y había que repartir los recursos informáticos (tiempo de procesador y memoria) entre varios usuarios.

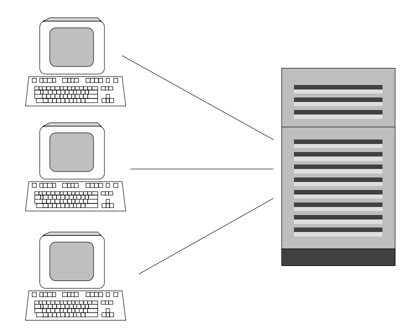


Figura 5.1. Terminales conectadas a un sistema central (*mainframe*): años sesenta

Los usuarios empleaban terminales compuestas por un monitor y un teclado para interactuar a distancia con el ordenador central. El teclado servía para enviar órdenes y el monitor para mostrar información, pero en realidad los programas se ejecutaban en el ordenador central. El término "tonto" que hemos empleado significa que en los terminales no se efectuaba ningún trabajo, ya que carecían de procesador. Este sistema de terminales remotas proporcionaba una serie de ventajas:

- Abarataba los costes totales del sistema informático, ya que un terminal "tonto" era mucho más barato y permitía el acceso remoto a los recursos.
- Permitía compartir varios recursos físicos, no sólo el procesador y la memoria del ordenador central, sino también impresoras, unidades de cinta y discos duros.
- Permitía compartir programas y datos, igualmente, ya que el ordenador central albergaba los datos con los que trabajaba el *software* instalado. Como hemos comentado, en realidad los terminales no ejecutaban los programas, ya que tan sólo mostraban los resultados en pantalla y permitían introducir datos por teclado de manera interactiva.
- Se abarataba el mantenimiento, ya que sólo había un único ordenador en el que instalar programas, del que hacer copias de seguridad, que reparar (cuando se estropeaba un terminal se tiraba directamente, dado su bajo coste), etc.

Si bien estos sistemas se pueden considerar como los precursores de las redes informáticas, fue una investigación patrocinada por el *Departamento de Defensa* de los EE.UU. (en adelante DoD, del inglés *Department of Defense*, para abreviar) a mediados de los sesenta la que dio origen a la red ARPANET, la cual se considera por varios autores como la antecesora de la red mundial de ordenadores conocida por el nombre de Internet.

A continuación, vamos a comentar cuáles fueron las ideas que impulsaron esta investigación, ya que, como buenos universitarios, debemos buscar y difundir la verdad. Viene esto a cuento porque por ahí circula la "leyenda urbana" de que Internet nació como consecuencia de un proyecto militar que uniera los centros de computación del DoD y esto no es cierto.

La leyenda cuenta que el DoD quería tener una red propia de ordenadores, pues estando en plena guerra fría, se buscaba un modo de comunicación que fuera resistente

a caídas repentinas de algunos de sus componentes (caídas que podían deberse a un ataque nuclear, es decir, averías graves). Sin embargo, en realidad, Internet surgió de la necesidad cada vez más acuciante de poner a disposición de los contratistas de la Oficina para las Tecnologías de Procesado de la Información (emplearemos el acrónimo IPTO del inglés original en adelante) cada vez más y más recursos informáticos.

El objetivo de la IPTO era, textualmente, *buscar mejores maneras de usar los ordenadores* yendo más allá de su uso inicial como grandes máquinas calculadoras. Pero, con el auge de los sistemas telemáticos, la IPTO, que era la encargada de distribuir los recursos informáticos entre centros de investigación y de cálculo, se enfrentaba al problema de que cada centro investigador y laboratorio que trabajaba para ella quería tener su propio ordenador. Esto ocasionaba numerosos problemas económicos y de logística: duplicidad de esfuerzos, coste elevado, más averías, etc. La IPTO tenía un presupuesto limitado y una serie de ordenadores para poner a disposición de los centros investigadores.

Entonces, Robert Taylor, a la sazón director de la IPTO, tuvo una brillante idea basada, a su vez, en las ideas de J. C. R. Licklider, plasmadas en los artículos *"Man-Computer Symbiosis"* y *"The Computer as a Communication Device"*, donde se introduce la idea de la comunicación de datos mediante conmutación de paquetes: *¿por qué no conectar todos los ordenadores de la IPTO entre sí?*. Al construir una serie de enlaces electrónicos entre diferentes computadores, los investigadores que estuvieran haciendo un trabajo similar en diferentes lugares del país podrían compartir recursos y resultados más fácilmente.

Eso tenía, además, un beneficio directo para la Agencia para Proyectos de Investigación Avanzados (en adelante emplearemos para abreviar su acrónimo inglés, ARPA por *Advanced Research Projects Agency*), de la cual dependía la IPTO de Taylor. En lugar de gastar el dinero en media docena de caros ordenadores distribuidos por todo el país, ARPA podría concentrar sus recursos en un par de lugares instalando allí ordenadores muy potentes y crear una red de comunicaciones en la que todo el mundo pudiera acceder a ellos.

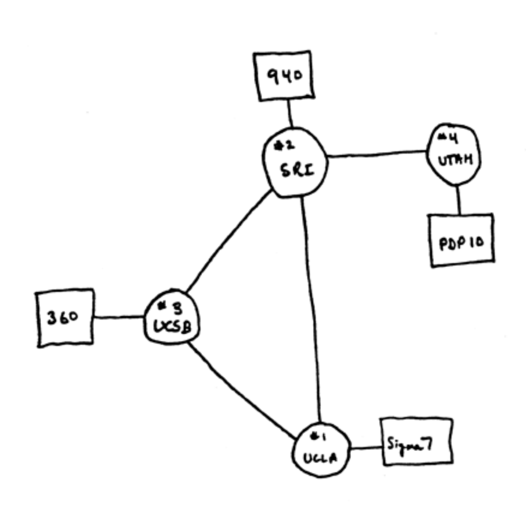


Figura 5.2. Ilustración original manuscrita (dibujada sobre un papel) de la red ARPANET

ARPA era la agencia encargada de la investigación militar y dependía del DoD estadounidense. Así, en 1965, esta agencia patrocinó un estudio acerca de "redes cooperativas de ordenadores de tiempo compartido". Como prueba de concepto, se conectaron entre sí el Lincoln Lab del MIT (*Massachussets Institute of Technology*) y el *System Development Corporation* de California mediante una línea dedicada que funcionaba a unos 1.200 bps. (bits por segundo). Compárese esta velocidad con las que obtenemos hoy en día mediante una conexión casera.

Con estos antecedentes, ARPA decidió asignar un millón de dólares al proyecto de Taylor. Y así es como en 1966 se

lanzó el primer plan ARPANET, auspiciado por Robert Taylor, director de la IPTO, dependiente de ARPA que, a su vez, dependía del DoD. Sin duda, esta cadena de dependencias es la que ha contribuido a dar pábulo a la leyenda que relaciona el origen de Internet con la guerra fría y la posibilidad de ataques nucleares.

El primer plan ARPANET tenía como objetivo principal el diseño de una red de comunicaciones *tolerante a la destrucción de algunos de sus nodos*. Esto, en realidad, era una idea de Taylor: si un ordenador de la red fallaba, los demás podrían seguir trabajando, lo que redundaría en mayor disponibilidad de los limitados recursos. Para ello, contaba con la tecnología denominada *conmutación de paquetes*, prevista por Licklider. La idea básica de la conmutación de paquetes consiste en *trocear* los mensajes (datos) que se envían en pequeños paquetes de datos denominados *datagramas*. Luego, cada paquete puede seguir una ruta diferente hasta llegar a su destino. Tiene que existir un *software* que se encargue de trocear la información en el ordenador origen, posteriormente enviar los datagramas por la línea de comunicaciones, y, finalmente, ordenar y juntar los paquetes de datos en el ordenador destino para recomponer la información originalmente enviada. Éste es el papel de lo que posteriormente sería el protocolo TCP (junto con el protocolo IP, su complementario de bajo nivel). La red ARPANET sería la precursora de lo que en la actualidad es la red mundial de ordenadores popularmente conocida como Internet.

Mientras ARPA estudiaba el problema y ponía en marcha el plan ARPANET, en el año 1968 comenzaba a funcionar la primera red basada en la tecnología de conmutación de paquetes. Ésta fue desarrollada en el NPL (*National Physical Laboratory*) en Middlesex, Inglaterra. Casi simultáneamente, en agosto de 1968, ARPA convocaba un concurso de proyectos para el desarrollo de los principales componentes de ARPANET. Los proyectos se recibieron en septiembre:

- La Universidad de California Los Ángeles (UCLA, por *University of California Los Angeles*) obtuvo en octubre el contrato para la construcción del denominado "Centro de mediciones de redes".

- La empresa *Bolt Beranek & Newman, Inc.* (BBN) ganó el contrato para construir los denominados Procesadores de Mensajes de Interfaz (IMP, acrónimo del inglés *Interface Message Processor*) y recibió un millón de dólares para el desarrollo del proyecto. Los IMP constituían un elemento clave en el desarrollo de ARPANET, ya que eran los nodos encargados de la conmutación y reenvío de los paquetes y se usaban para conectar los computadores a la red ARPANET. Fueron, de hecho, la primera generación de los dispositivos que hoy conocemos con el nombre de *enrutadores* o *routers*.

La empresa BBN cumplió con su cometido y desarrolló la infraestructura de la red basada en la utilización de los IMP. Sin embargo, no se hizo cargo del desarrollo del *software* necesario para el funcionamiento de la misma. Para realizar los programas, se hizo una reunión de investigadores en redes (muchos de ellos estudiantes de doctorado). Como resultado de esta reunión se creó el *Network Working Group* (NWG), grupo de trabajo encargado de desarrollar y preparar los protocolos para la comunicación en la red ARPANET. En diciembre de 1969 comenzó a funcionar ARPANET con tan sólo cuatro nodos. Ellos fueron los "originales":

1.- Universidad de California Los Ángeles (UCLA).

2.- Instituto de Investigación de Stanford (SRI, del inglés *Stanford Research Institute*).

3.- Universidad de California Santa Bárbara (UCSB).

4.- Universidad de Utah.

En 1970 se publica el "Host-to-host protocol" que describe el funcionamiento de un protocolo de comunicaciones de computador a computador. Los computadores conectados a la red ARPANET comienzan a usar el denominado *Network Control Protocol* (NCP), el primer protocolo de comunicaciones "host-to-host" basado en el trabajo anterior. En 1971, la red ARPANET contaba ya con 15 nodos: a los originales UCLA, SRI, UCSB y Universidad de Utah, se les añaden BBN, MIT, RAND (*Research ANd Development Corporation*), SDC (*System Development Corporation*), Universidad de Harvard, Lincoln Lab, Universidad de Stanford, UIUC (*University of Illinois at Urbana-Champaign*), CWRU (*Case Western Reserve University*), CMU (*Carnegie Mellon University*) y NASA Ames Research Center (centro de investigación de la NASA en la frontera entre las ciudades de Mountain View y Sunnyvale en California).

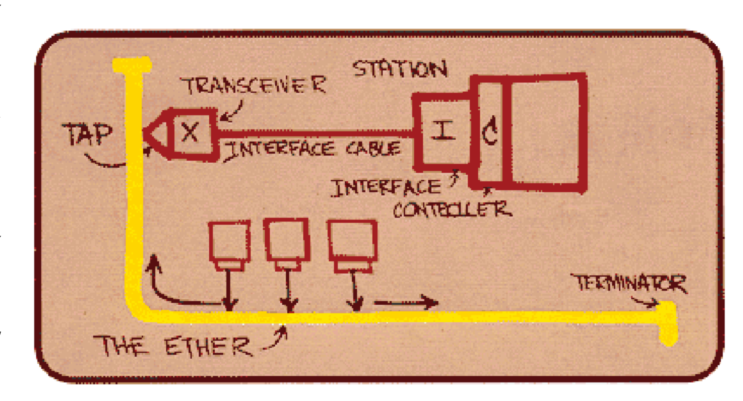


Figura 5.3. Diseño manuscrito de la red Ethernet de Robert Metcalfe

En aquel tiempo (año 1971) Raymond Tomlinson, un programador que trabajaba para la empresa BBN, diseñó el primer programa de *correo electrónico* capaz de enviar mensajes entre usuarios en diferentes ordenadores conectados a ARPANET. Previamente, el correo sólo podía enviarse entre los usuarios del *mismo* computador. Para lograrlo, creó el concepto de direcciones de correo electrónico y empleó el signo @ para separar el nombre (identificador) de usuario de la dirección de la máquina en la que éste tenía la cuenta. El signo @ fue escogido por el propio Ray Tomlinson a partir de las teclas de su teclado teletipo Modelo 33. En 1972, este programa se adaptó a ARPANET y Lawrence Roberts se encargó de diseñar un programa "cliente" para este servicio de correo electrónico que permitía listar correos, archivarlos, responder, etc.

Por esa época, también se integra en ARPANET el protocolo (servicio de conexión remota) *telnet* que fue originalmente desarrollado en el año 1969. Poco después, en el año 1973, se produce otro hito importante en la historia de las redes de ordenadores: Robert Metcalfe pergeña lo que será posteriormente *Ethernet*. Bob Metcalfe estaba trabajando entonces en Xerox PARC (acrónimo del inglés *Palo Alto Research Center*) y su desarrollo de Ethernet fue compartido con David Boggs. Ethernet fue originalmente un estándar para la conexión de computadores en cortas distancias (unos pocos centenares de metros), pero en la actualidad ha evolucionado hasta convertirse en un

estándar para la conexión de redes de área local empleando normalmente una topología en estrella. La red originalmente desarrollada por Metcalfe y Boggs se probó por primera vez en el propio centro de investigación Xerox PARC (*Alto Ethernet*).

En el año 1974, la empresa BBN ofreció el primer servicio público de datos, es decir, lo que podría considerarse como la versión comercial de ARPANET. A partir de entonces, ARPANET creció rápidamente y comenzó a cubrir buena parte de Estados Unidos. Al crecer ésta, se vio que los protocolos inicialmente desarrollados para la gestión de las comunicaciones no eran del todo adecuados. Por entonces, Vint Cerf y Robert Kahn publican un artículo científico titulado *"A Protocol for Packet Network Interconnection"*, que describe el protocolo de comunicaciones denominado *Programa de Control de Transmisiones*, el cual se corresponde con las siglas del acrónimo inglés TCP. Éste sería el protocolo de comunicaciones que definitivamente se adoptó en ARPANET. Con el fin de aumentar la popularidad del TCP, ARPA contrató a la universidad de Berkeley para que lo integrara en su UNIX. En esta universidad también se desarrollaron gran cantidad de utilidades y aplicaciones para la red, como, por ejemplo, el servicio de FTP (*File Transfer Protocol*) para transferencia remota de ficheros.

Los primeros enlaces intercontinentales vía satélite aparecieron en el año 1975. Por cuestiones de operatividad y eficiencia, en el año 1978 se produjo una revisión importante del protocolo TCP, núcleo de las comunicaciones en ARPANET. Éste se dividió en dos partes: TCP (*Transmission Control Protocol*), encargado de la transmisión fiable de la información *como un todo* en redes que, en principio, se asume que no son fiables, e IP (*Internet Protocol*), que se encarga del envío y recepción de los datagramas a través del medio físico correspondiente. Precisamente, las *direcciones IP* son los números que sirven para identificar cada máquina en la red y que se emplean para el envío de los paquetes de datos.

En el año 1979 se reunieron la Universidad de Winsconsin, ARPA, la Fundación Nacional de Ciencia de los EE.UU. (NSF, del inglés *National Science Foundation*) e investigadores de varias universidades con el fin de estudiar la creación de una red de computadores que uniese diversos centros de investigación públicos y departamentos de informática. En ese mismo año, aparece USENET (contracción del inglés *USEr NETwork*). USENET es un sistema de discusión distribuido, global y descentralizado. Fue concebido por Tom Truscott y Jim Ellis. El sistema permite que los usuarios lean y publiquen mensajes públicos (artículos o *posts*), colectivamente denominados *news*, y agrupados en varias categorías o *grupos de noticias* (*newsgroups*). USENET recuerda en muchos aspectos de su estructura a los antiguos sistemas BBS (*Bulletin Board Systems*) y fue el precursor de los foros, ampliamente usados en la web hoy en día. En USENET las discusiones sobre un mismo asunto estaban enlazadas en un mismo "hilo" (*thread*), aunque los mensajes se almacenaban secuencialmente en el servidor.

También por esas fechas (años 1978 y 1979) se diseñaron los primeros MUD (acrónimo del inglés *Multi-User Dungeon, Domain o Dimension*), que eran juegos de aventuras interactivos multiusuario en modo texto. Poco después, en el año 1982, Scott Fahlman propone el uso de "emoticonos" (*emoticons* en inglés, también conocidos popularmente como *smileys*) en los mensajes de correo electrónico para discernir las bromas del texto serio y poder expresar, en cierta medida, emociones. Los primeros fueron `:-)` y `:-(`.

Otro hito histórico importante lo constituye el famoso bloqueo de ARPANET ocu-

rrido el 27 de octubre de 1980 a causa de un virus informático que se propagó de manera accidental. En aquella época aproximadamente veinte mil personas tenían acceso a esta red a través de los ordenadores de unos 200 centros gubernamentales de investigación, universidades y centros militares. Este incidente mostró la importancia que tenía la seguridad en la red. En los años 1981 y 1982 surgen diversas redes con diferentes propósitos:

- BITNET (acrónimo del inglés *Because It's Time NETwork*) fue el resultado de la cooperación entre la universidad de la ciudad de Nueva York y la Universidad de Yale. Ofrecía servicio de correo electrónico, listas de correo y transferencia de ficheros.
- CSNET (del inglés *Computer Science NETwork*), financiada por la estadounidense Fundación Nacional de Ciencia (NSF), fue una red de computadores que enlazaba departamentos de informática de varias universidades esparcidas por todo el país, entre ellas la Universidad de Delaware, la Universidad de Purdue y la Universidad de Winsconsin. También colaboraron con el proyecto RAND Corporation y BBN. El objetivo que se perseguía era ser una alternativa a ARPANET, proporcionando acceso a departamentos de informática que no tenían el privilegio de acceder a ella. CSNET se conectaba con ARPANET mediante TCP/IP, pero también soportaba conexiones vía módem telefónico para aquellos departamentos que tenían medios más modestos. Fue la precursora de la posterior NSFNET.
- Minitel era un sistema francés de acceso telefónico a datos mediante un terminal doméstico. Se trataba de un servicio "en línea" (*online*) a través del teléfono, considerado como uno de los más exitosos antes de la llegada de *World Wide Web* (WWW). Sus usuarios podían hacer compras *online*, reservas de billetes de tren, comprobar la cotización de las acciones en la Bolsa, buscar en el listín telefónico y charlar (*chat*) de la misma forma que hoy es posible con Internet y la WWW.
- EUNET (acrónimo del inglés *European UNIX NETwork*) fue el fruto de la colaboración entre entidades individuales pertenecientes a Holanda, Dinamarca, Suecia y Reino Unido. Ofrecía servicios de correo electrónico y noticias (USENET).

En 1983 se desarrolló un *servidor de nombres* en la Universidad de Winsconsin. Éste permitía que los usuarios de la red pudieran enviar mensajes sin conocer la ruta exacta que tenían que seguir. También se creó una *pasarela* que servía de interconexión entre las redes CSNET y ARPANET. En ese mismo año se decidió dividir ARPANET en dos inter-redes: ARPANET propiamente dicha, que contaba con 45 nodos, y MILNET (del inglés *MILitary NET*), que contaba originalmente con 68 nodos. MILNET fue diseñada para gestionar el tráfico no clasificado del DoD y se creó fundamentalmente por razones de seguridad. ARPANET permaneció para dar servicio a la comunidad científica universitaria, aunque la conectividad entre ambas redes era bastante restringida. Ambas redes usaban una tecnología similar y se emplearon pasarelas para enviar el correo electrónico entre ellas.

Entre los años 1983 y 1984, aparecieron las primeras *estaciones de trabajo* UNIX. Eran potentes computadores (para la época) dedicados a cálculo científico y computación de altas prestaciones que llevaban instalado el sistema operativo UNIX desarrollado por la Universidad de Berkeley (UNIX BSD 4.2), el cual incluía el *software* de interconexión TCP/IP de forma nativa, junto con muchas utilidades y aplicaciones para la red, como programas de correo electrónico, conexión remota (telnet) y transferencia de ficheros (FTP).

En general, durante los ochenta, muchas redes se fueron uniendo a ARPANET. Una de las más importantes de estas redes fue CSNET que, como hemos comentado anteriormente, fue dirigida y promovida por la *National Science Foundation* (NSF), agencia gubernamental estadounidense para la promoción de la ciencia. CSNET fue el origen de una red mucho mayor, la NSFNET (*National Science Foundation Network*). Esta red surgió a mediados de los ochenta, tras el abandono de CSNET. La NSF pretendía crear una red abierta que permitiese la utilización de grandes supercomputadores por parte de los investigadores universitarios.

En 1985, la NSF comenzó a financiar la creación de cinco centros de superordenadores en las universidades más grandes: el Centro de Investigación John von Neumann en la Universidad de Princeton, el Centro de Supercomputación de San Diego en el campus de la Universidad de California en San Diego (UCSD), el Centro Nacional para Aplicaciones de Supercomputación en la Universidad de Illinois en Urbana-Champaign (UIUC), el Centro de Teoría de Cornell en la Universidad de Cornell y el Centro de Supercomputación de Pittsburgh. Sólo fueron cinco debido a su elevado coste, lo cual implicaba que debían compartirse por el resto de universidades y centros de investigación. Ello creó un problema de comunicaciones, ya que se necesitaba conectar entre sí los centros de supercomputación y permitir a los clientes de estos centros (que pertenecían a otras redes) acceder a ellos.

Debido a problemas burocráticos y administrativos, la NSF no pudo emplear ARPANET para este propósito, por lo que decidió construir su propia red basándose en la tecnología TCP/IP de ARPANET. Sin embargo, resultaba inviable conectar cada universidad directamente con un centro de supercomputación, por el gasto y por el posible problema de colapso del centro en cuestión. Por ello, se decidió crear *redes regionales*: en cada área del país, las universidades y centros de investigación podrían conectarse con su vecino más próximo. Cada grupo, universidad o centro de investigación fue conectado a un centro de supercomputadores en un punto.

La red de la NSF (NSFNET) conectó estos cinco centros entre sí y permitió, de esta forma, el acceso a los supercomputadores en toda la red sin coste alguno. Con esta forma de conexión, cualquier ordenador puede comunicarse finalmente con cualquier otro llevando la conversación a través de sus vecinos.

En 1986, NSFNET comenzó a funcionar usando un protocolo propio basado en el TCP/IP que era compatible con ARPANET. NSFNET constituía la espina dorsal a través de la cual se conectaban redes regionales y académicas. En poco tiempo, experimentó un crecimiento exponencial en su tráfico: los enlaces originales que funcionaban a 56 Kbps. (kilobits por segundo) fueron actualizados por otros que iban a $1{,}5$ Mbps. (megabits por segundo) en 1988, y de nuevo en 1991, por otros que funcionaban a 45 Mbps.

Por otro lado, surgieron grandes avances en el terreno de las *redes locales* y en la

informática personal: se abarataron los ordenadores personales y las estaciones de trabajo, y comenzaron a aparecer terminales “inteligentes” (ordenadores personales con procesador, disco y memoria propia) como componentes a conectar a las redes; ¡estamos a mediados de los *años ochenta*!. En lo que se refiere a las redes, los avances se realizaron sobre todo en el desarrollo de los equipos de conexión: mejores y sofisticados *routers* trajeron consigo nuevas maneras de interconexión entre los equipos de la red (topologías, como por ejemplo en estrella).

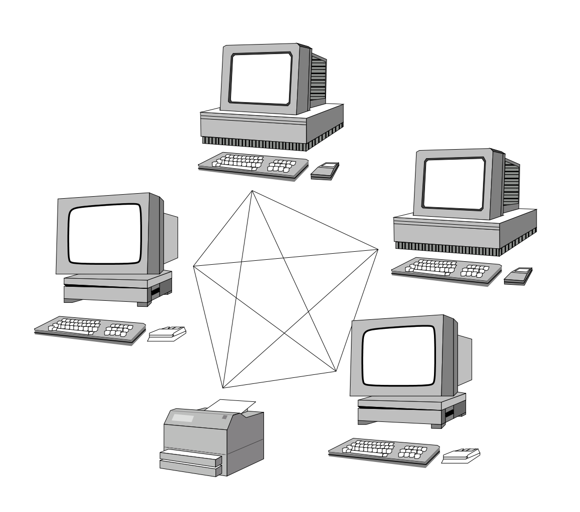


Figura 5.4. Ordenadores “inteligentes” se conectan entre sí y se comparten los recursos en red: años ochenta

El hasta entonces “ordenador central” pasó a ser un *servidor*: era más barato que el *mainframe* central, pues se conectaba a otros ordenadores “inteligentes” que, al poseer su propio procesador, disco y memoria, descargaban al servidor de trabajo y posibilitaban que éste fuese más “modesto” en cuanto a prestaciones. En ocasiones no había un único servidor: todas o muchas de las máquinas de la red servían ficheros y otros recursos. Esto hizo que el sistema informático, considerado globalmente, fuese más fiable y tolerante a fallos y aumentase la capacidad de cálculo a medida que crecía la red incorporando nuevos equipos. Además, al existir varios tipos de ordenadores con capacidad de conexión a una red local, el equipamiento informático se podía adaptar a las necesidades de cada puesto de trabajo: estaciones gráficas, ordenadores de bajas prestaciones, sistemas con *hardware* específico, etc. dotan de mayor flexibilidad a los sistemas informáticos de las empresas y corporaciones.

Muchas de las redes locales que iban surgiendo se crearon siguiendo el diseño de NSFNET que hemos descrito en párrafos anteriores. Así, tras una etapa de gran crecimiento, se hizo necesario la adopción de un sistema para almacenar los números o direcciones IP de los ordenadores. Para solucionar este problema, se introdujo en el año 1984 el servicio para el “Sistema de Nombres de Dominio”, más conocido como DNS (acrónimo del inglés *Domain Name System*), basándose en los trabajos realizados en el año 1983 en la Universidad de Winsconsin. El servicio de DNS permite, básicamente, asociar un nombre a cada ordenador conectado a la red y emplear éste en lugar de su dirección IP para realizar la conexión y transferencia de datos. En el sistema DNS, los nombres se estructuran jerárquicamente mediante un sistema de dominios, del que comentaremos más cosas en posteriores secciones de este capítulo. Como resultado de estos avances, se pasó rápidamente a tener más de 1.000 nodos conectados a la red.

También en 1986, concretamente en enero, se fundaron la IETF (del inglés *Internet Engineering Task Force*) y la IRTF (del inglés *Internet Research Task Force*). La IETF (`http://www.ietf.org/`) es, en la actualidad, una organización que promueve y desarrolla los estándares en Internet, cooperando estrechamente con el *W3 Consortium* (W3C), encargado del establecimiento de estándares para el servicio Web

(WWW) como, por ejemplo, el HTML[1], y con ISO e IEC (*International Organization for Standardization* e *International Electrotechnical Commission*, respectivamente), organizaciones para la creación y mantenimiento de estándares. En particular, la IETF tiene que ver con estándares relacionados con TCP/IP y el conjunto de protocolos usados en Internet.

¿Sabías que. . . ?

La IETF es, igualmente, una organización de estándares abiertos, sin una membresía formal o requisitos específicos para ello. Todos los participantes y líderes son voluntarios, aunque su trabajo está normalmente financiado por sus empresas o patrocinadores. Por ejemplo, el actual director está financiado por la empresa *VeriSign* y la Agencia Nacional de Seguridad (NSA) estadounidense.

Esta organización está compuesta por un gran número de grupos de trabajo y de grupos de discusión (informales). Cada grupo trata de completar un trabajo sobre un determinado tema de interés y después se disuelve.

Cada grupo de trabajo tiene un presidente o director que es designado previamente (a veces hay más de un director o co-directores). También, debe existir un documento que describe cuál es el objetivo del grupo de trabajo y qué y cuándo se espera producirlo.

Los diferentes grupos de trabajo se organizan en áreas según los temas y asuntos generales que se abordan en la IETF. En la actualidad, existen las áreas de *Aplicaciones*, *General*, *Internet*, *Operaciones y Dirección*, *Aplicaciones en tiempo real* e *Infraestructura, Encaminamiento, Seguridad y Transporte*.

Cada una de estas áreas es supervisada por un director o gerente de área, aunque en la mayoría de las áreas se suelen emplear dos codirectores. Éstos son responsables de elegir a los presidentes de los grupos de trabajo de su área.

Los directores de área junto con el director o presidente de la IETF forman el IESG (*Internet Engineering Steering Group*), encargado de gestionar el funcionamiento general de la IETF.

Por su parte, la IRTF (`http://www.irtf.org/`) es un organismo estrechamente relacionado con la IETF y está compuesto por grupos de investigación que estudian a largo plazo asuntos que tienen que ver con Internet y diversas tecnologías relacionadas.

En 1987, Gordon Bell, que por entonces ocupaba el cargo de jefe del Directorio de Computación de la NSF, lideró el grupo que ideó y proyectó una red de educación e investigación como respuesta a una petición del entonces congresista Al Gore. Se trataba de la Red Nacional de Investigación y Educación, NREN (acrónimo del inglés *National Research and Education Network*). Por aquella época habían ya unos 10.000 nodos conectados en la red.

[1]Éste es el “lenguaje” con el que están hechas las páginas web y que deben interpretar los navegadores.

Propósito de la IRTF

Según reza en su página web, la misión de la IRTF es "promover investigaciones de importancia para la evolución de la futura Internet mediante la creación de pequeños grupos de investigación a largo plazo, centrados en trabajar con temas relacionados con protocolos, aplicaciones, arquitectura y tecnología en Internet".

El dos de noviembre de 1988, Robert Morris Jr., un estudiante de informática en la Universidad de Cornell, escribió un programa experimental, capaz de replicarse a sí mismo y de propagarse por la red. Este programa era un *gusano* que Morris lanzó en lo que entonces era Internet. Realmente, lo lanzó desde el MIT para disimular el hecho de que el gusano venía en realidad de Cornell. Para su sorpresa, Morris descubrió que el programa se copiaba y, por tanto, infectaba computadores a un ritmo mucho más rápido de lo que él suponía debido a un fallo en el código.

Al fin, muchos ordenadores en diferentes lugares del país, incluyendo universidades, puestos militares y centros de investigación médica, se "colgaron" o bien quedaron en estado "catatónico". El célebre *gusano de Internet* afectó a 6.000 de los 60.000 nodos que por entonces estaban conectados. Se estima que el coste de "combatir" a este gusano en cada una de las instalaciones afectadas varió entre los 200 y los más de 53.000 dólares de la época.

Curiosidades acerca del gusano de Morris

Cuando Morris se dio cuenta de lo que estaba sucediendo, se puso en contacto con un amigo de la Universidad de Harvard para discutir con él una posible solución. Parece ser que enviaron un mensaje anónimo por la red desde Harvard, indicando a los programadores cómo podían "matar" al gusano y prevenir la infección en los ordenadores "sanos". Sin embargo, y puesto que la red estaba saturada por el efecto del propio gusano, el mensaje no llegó a su destino hasta que fue demasiado tarde.

El gusano de Morris se aprovechaba de un fallo en el modo de depuración del programa `sendmail` de UNIX (un servidor de correo), el cual se ejecutaba en un sistema y esperaba a que otros sistemas se conectasen a él y le entregasen el correo. También sacaba partido de un fallo en el "demonio" `fingerd` –los demonios son programas que están siempre ejecutándose en segundo plano (no somos conscientes de su actividad)– en los sistemas UNIX, el cual responde a peticiones de consulta de datos acerca de los usuarios del sistema.

Programadores de la Universidad de California en Berkeley y del MIT consiguieron copias del gusano y estuvieron intentando obtener el código fuente a partir del binario (ingeniería inversa o "desmontaje") para intentar averiguar cómo funcionaba y así poder detenerlo.

Equipos de programadores trabajaron sin descanso para tratar de encontrar al menos un parche temporal y así evitar la continua difusión del gusano.

continúa en la página siguiente ...

. . . viene de la página anterior

Después de 12 horas, el equipo de Berkeley logró establecer una secuencia de pasos que ayudaría a retardar la difusión del virus. Otro método fue descubierto en la Universidad de Purdue y fue ampliamente divulgado. No obstante, la información no llegó todo lo rápido que podría ya que muchos sitios se habían desconectado completamente de la red dado el colapso existente.

Tras unos días, la situación lentamente volvió a la normalidad y todo el mundo quería averiguar quién había sido el autor de todo este embrollo.

El periódico *New York Times* mencionó que Morris era el autor, aunque esto todavía no había sido oficialmente probado, ya que había bastantes evidencias apuntando en su contra.

Robert Morris Jr. fue condenado por violar el Acta sobre Fraude y Abuso con computadores en su título 18, y fue sentenciado a 3 años de supervisión y vigilancia (no supone ingreso en prisión), 400 horas de trabajo comunitario y una multa de 10.050 $. Su apelación, hecha en diciembre de 1990, fue rechazada 3 meses después.

La creación de la Autoridad de Números Asignados en Internet, IANA, acrónimo del inglés *Internet Assigned Numbers Authority* supuso otro evento importante, que ocurrió en 1988. Ésta es la entidad que aún hoy en día se encarga de supervisar la asignación de direcciones IP a nivel global para todos aquellos ordenadores que se conectan a Internet. Además, se encarga también de gestionar los sistemas DNS primarios (o DNS raíz) y de las asignaciones para otros protocolos de Internet menos conocidos que TCP/IP. En la actualidad, la gestión de IANA es mantenida por la ICANN (*Internet Corporation for Assigned Names and Numbers*). En el año 1989 ya había en torno a 100.000 nodos conectados a la red.

Sobre IANA

IANA es responsable a nivel mundial de la asignación de nombres y direcciones IP *únicos*, ya que no puede existir más de una máquina conectada directamente a Internet con el mismo nombre (de dominio) o dirección IP. Lo mismo debe cumplir con las asignaciones referidas a otros protocolos públicos en Internet. Por ello, mantiene una estrecha relación con la IETF.

Cuando se trata de administrar subconjuntos de "alto nivel", como por ejemplo direcciones IP en nombres de dominio, se utilizan métodos de supervisión extraordinarios y más específicos, ya que, puesto que los administradores de cada dominio gestionan su rango de direcciones IP, hay que tratar con un sistema de administración con múltiples capas.

Como hemos visto, a lo largo de los años ochenta hubo una auténtica explosión en el terreno de las redes locales. Muchas de estas redes se unieron directa o indirectamente a ARPANET. Además, por cuestiones de seguridad, ésta había sido separada de la red militar propiamente dicha (MILNET), perdiendo buena parte de su estructura. To-

do ello contribuyó a que, finalmente, ARPANET fuese desmantelada y desapareciese como tal definitivamente en el año 1990.

La unión de lo que fue ARPANET, NSFNET, NREN y las demás redes comenzó a llamarse Internet y se extendió a gran cantidad de países con un crecimiento exponencial (se estima que el número de ordenadores conectados se duplica aproximadamente cada año). En el año 1991, Sir Tim Berners-Lee, un científico que por entonces trabajaba en la Organización Europea para la Investigación Nuclear, CERN (acrónimo del francés *Conseil Européen pour la Recherche Nucléaire*), ideó y desarrolló el servicio tal vez más popular de Internet[2]: *World Wide Web*, más conocido como WWW. Utilizamos el tratamiento de "Sir" para referirnos a Berners-Lee porque fue nombrado Miembro de la Orden de Mérito por la Reina Isabel II de Inglaterra en junio de 2007. Sin duda, éste ha sido el servicio que más ha contribuido a la popularización y el uso masivo de la red por parte de los neófitos.

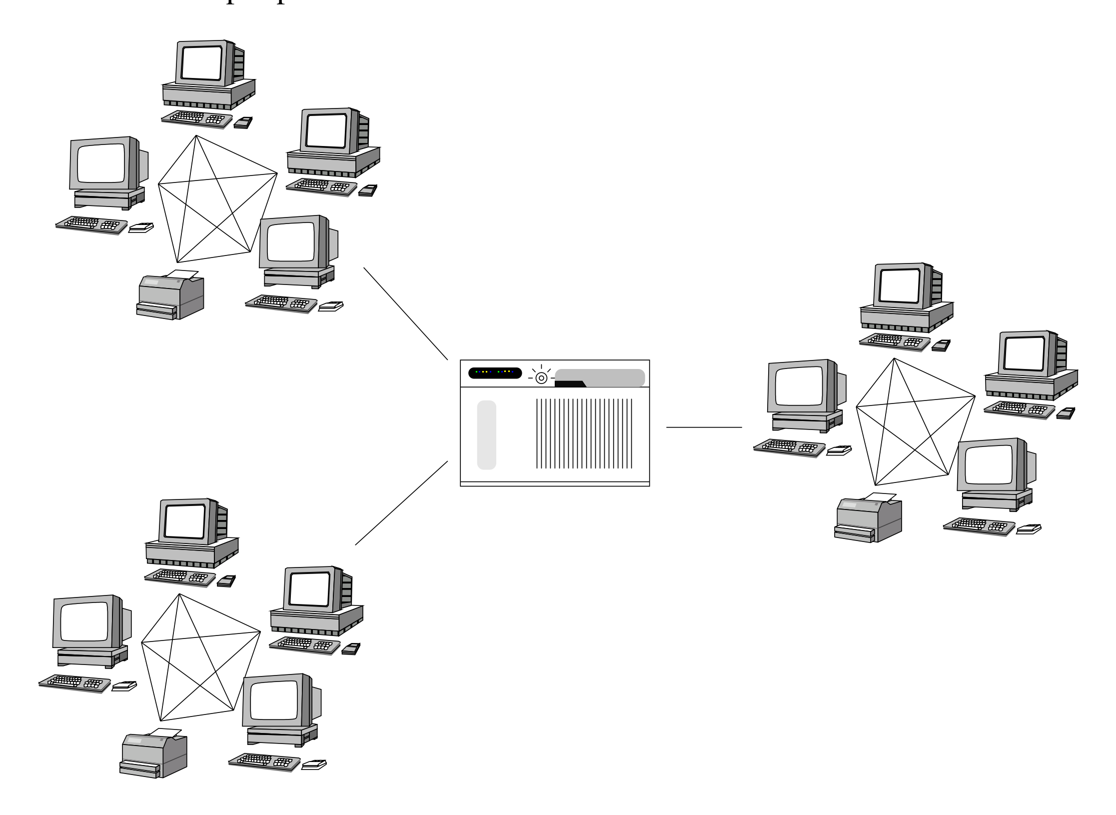


Figura 5.5. Redes locales, metropolitanas y de área ancha pasan a interconectarse entre sí posibilitando un crecimiento enorme de Internet: años noventa

En los años noventa se pasa de interconectar entre sí grandes computadores de tiempo compartido (multiusuario) a interconectar redes locales de computadores enteras mediante la utilización de pasarelas, puentes, conmutadores (*switches*) y encaminadores (*routers*) más avanzados (véase la Figura 5.5). Esto posibilita el *gran auge* de Internet: ésta llega al gran público y su uso crece rápidamente entre la gente, debido sobre todo a la facilidad de uso y popularidad de servicios como el WWW (mediante la utilización de navegadores) y, más recientemente, de las denominadas "redes de pares" (P2P, del inglés *peer-to-peer*). Como consecuencia de todo ello, ocurre que:

[2]Con permiso de las redes de pares (P2P).

- Se comunican entre sí departamentos lejanos (empresas multinacionales y nacionales) con bajo coste.
- Surgen estándares de interconexión que simplifican y abaratan el diseño y la construcción de las redes informáticas (conmutadores o *switches* programables, por ejemplo).
- La red es altamente configurable: se pueden cerrar determinados puertos al tráfico a voluntad, se puede dar prioridad a determinados paquetes frente a otros (prioridad por servicios), etc.
- Se puede compartir el uso de virtualmente cualquier dispositivo (impresoras, discos, trazadores, etc.) capaz de ser conectado a un ordenador.
- Aumenta enormemente el *impacto social* de la tecnología informática. Las redes de computadores están presentes en muchísimos ámbitos: empresa, educación, ocio, hogar...

En el año 1992 se alcanza ya el millón de nodos conectados a Internet. También se publica una obra clásica: *"Zen y el Arte de Internet"*, cuyos contenidos todavía se pueden consultar en `http://www.cs.indiana.edu/docproject/zen/zen-1.0_toc.html`. Aunque está ya algo anticuado, sigue siendo interesante para comprender qué es y cómo funciona Internet, y también algunas de sus herramientas básicas. Recomendamos la lectura de algunos de sus pasajes.

Para aprovechar el desarrollo ya mencionado del servicio WWW, aparece en el año 1993 el primer navegador web: se trata del popular *Mosaic*[3]. Con la llegada del servicio web se popularizan los buscadores de páginas (año 1995). Por esas fechas, aparece el lenguaje de programación Java, íntimamente ligado al desarrollo de aplicaciones informáticas (programas) para WWW. A principios de este siglo XXI, contamos ya con casi *ciento cincuenta millones* de nodos conectados a la red y con más de *treinta y seis millones* de sitios web en Internet (enero de 2002).

5.3. TRANSMISIÓN DE DATOS

Las redes de ordenadores suelen ser objetos complejos. Por ello, se divide su diseño en distintas capas con funciones bien diferenciadas. En este capítulo seguiremos la estructura que se describe en el Cuadro 5.1, la cual coincide *grosso modo* con la estructuración seguida en el denominado modelo de referencia de Interconexión de Sistemas Abiertos (OSI, del inglés *Open System Interconection*).

En el nivel físico, la preocupación es transmitir la información de un ordenador a otro cuando ambos están físicamente conectados mediante un soporte concreto. Para conectarse al medio o soporte determinado se suele emplear un aparato adaptador (por ejemplo, un módem sería el adaptador que emplearíamos para conectar un ordenador al soporte proporcionado por el cable telefónico). En muchos casos, se divide este nivel en dos partes:

[3] Usado por quien esto suscribe en sus tiempos predoctorales.

	Nivel
TELNET FTP SMTP DNS ...	Nivel de aplicación (cliente) o servicio
TCP UDP	Nivel de transporte
IP	Nivel de red
Soportes de red y adaptadores	Nivel físico

Cuadro 5.1. Estructura de niveles que vamos a utilizar como hilo conductor del capítulo

- El nivel físico en sentido estricto: se preocupa de los aspectos físicos del problema de la comunicación (qué voltaje se puede interpretar como un cero, cuál como un uno).
- El nivel de datos: estudia los algoritmos para conseguir una comunicación eficiente y razonablemente segura entre dos ordenadores adyacentes.

Nosotros no entraremos en tanto detalle, simplemente presentaremos algunos de los conceptos y problemas que se presentan en este nivel. En general, la forma de comunicación más sencilla entre dos dispositivos (que podemos interpretar como una "red") se compone de dos sistemas conectados capaces de enviarse datos entre sí (ordenadores, ordenador e impresora, etc.). En un ordenador podemos encontrar *hardware* específico para realizar la interconexión con otro ordenador o bien a un periférico para el envío y recepción de datos: se trata de los *puertos* (por ejemplo, serie, paralelo, USB...).

No obstante, nosotros estamos hablando de dispositivos especialmente diseñados para interconectar computadores de forma remota (o computadores y periféricos), formando lo que conocemos como redes de computadores. Para realizar este tipo de conexiones, necesitamos ocuparnos por un lado de la *conexión física* y, por el otro, de la forma en la que se *establece* y *realiza* la *comunicación*.

¿Sabías que...?

OSI, lanzado en 1984, fue el modelo de red descriptivo creado por ISO. El objetivo que se buscaba era proporcionar a los fabricantes un conjunto de estándares que aseguraran una mayor compatibilidad e interoperabilidad entre los distintos tipos de tecnología de red producidos por las empresas a nivel mundial.

Siguiendo el esquema de este modelo se crearon numerosos protocolos que durante muchos años ocuparon el centro de la escena de las comunicaciones informáticas. El advenimiento de protocolos más flexibles donde las capas no están tan demarcadas y la correspondencia con los niveles no era tan clara puso a este esquema en un segundo plano. Sin embargo, sigue siendo muy usado en la enseñanza como una manera de mostrar de qué manera puede estructurarse una "pila" de protocolos de comunicaciones.

En lo que se refiere a la conexión física hablaremos del soporte o medio físico que utilizamos para conectar los ordenadores, esto es, el *soporte de red*. Podría ser, por ejemplo, el cableado telefónico existente (par trenzado), o bien fibra óptica como emplean las redes de cable o incluso ondas de radio (Wi-fi). El soporte de red determina en cierta medida el *tipo de señal* usada en la transmisión y la necesidad de emplear determinados dispositivos adaptadores que sean capaces de tratarla. Por ejemplo, el cable telefónico convencional requiere que la señal que se envíe sea analógica; puesto que los ordenadores sólo saben interpretar señales digitales estamos obligados a usar un adaptador que sea capaz de convertir la señal analógica en digital y viceversa, o sea, un módem.

Muchos de los medios físicos empleados, por sus características, imponen una serie de restricciones en cuanto a la *distancia* máxima a la que es posible establecer la comunicación con garantías. Tal es el caso, por ejemplo, de la línea telefónica convencional, muy ruidosa y poco fiable. Igualmente, el medio físico utilizado determina la *velocidad* a la que se realizará la transmisión de datos: el canal que empleemos tendrá un ancho de banda o *bitrate* determinado, que puede ser o no compartido y que puede funcionar en ambos sentidos simultáneamente o no (*full duplex*). También cabe considerar que hay medios más *fiables* que otros (como hemos dicho, el cable telefónico convencional es poco fiable ya que se producen muchos errores en la transmisión de datos), de la misma forma que hay otros medios más *seguros* que otros: las ondas de radio que van por el aire son, por la naturaleza del medio en el que se transmiten, inherentemente inseguras, ya que cualquiera equipado con un receptor funcionando en la frecuencia adecuada puede interceptar las transmisiones. Obviamente, en este caso, se impone la utilización de un protocolo de comunicaciones que sea capaz de cifrar la información transmitida.

Todo lo concerniente a la forma en que se establece y realiza la conexión, esto es, las normas que definen cómo se realiza la comunicación, es lo que denominamos formalmente el *protocolo de comunicación*. Éste define cómo debe iniciar la conversación un ordenador determinado (*conexión*), es decir, qué tipo de paquetes y con qué datos debe transmitir. También se ocupa de definir estrictamente la forma en la que se enviarán y recibirán los datos, es decir, tanto el *formato* de los datos que se transmiten como la manera en la que se realizará el *reconocimiento* de la recepción de dichos datos (para, por ejemplo, retransmitirlos de nuevo si no se han recibido o transcurre un determinado lapso de tiempo sin recibir esta confirmación).

Igualmente, y puesto que los datos pueden *deteriorarse* durante la transmisión[4], es importante que el protocolo de comunicación sea capaz de *detectar* los fallos que puedan haber ocurrido durante la transmisión. Esto puede hacerse mediante la introducción, por ejemplo, de códigos de redundancia cíclica (CRC) u otros mecanismos más sofisticados de detección (*checksums*) y corrección de errores. Finalmente, el protocolo debe dar cuenta de la manera en la que debe cerrarse la comunicación entre dos partes, es decir, tiene qué definir la serie de acciones que se han de llevar a cabo para realizar la *desconexión*. Normalmente, toda esta faena no la realiza un único protocolo, sino que suele ser labor de varios operando de forma coordinada. Éste es el caso, por ejemplo, de los protocolos TCP e IP, tan acoplados en su funcionamiento que suelen

[4]Es relativamente fácil que un bit cambie su valor de 1 a 0 o viceversa.

referenciarse como un único protocolo denominado TCP/IP.

5.3.1. Tipos de señal

Básicamente, tenemos dos tipos de señales, cuya "forma" podemos apreciar en la Figura 5.6:

- *Señal analógica.* La señal analógica es aquélla que puede adoptar cualquier valor, dentro de unos límites, a lo largo del tiempo. Es una función continua en el tiempo y, aun estando limitada, puede tomar infinitos valores (piénsese, por ejemplo, que en el intervalo $[0, 1]$ de la recta real hay un número infinito *no numerable* de valores posibles).
- *Señal digital.* La señal digital es aquélla que sólo puede tomar valores dentro de un conjunto discreto conocido de antemano. Ese conjunto puede constar de un número infinito de valores, pero en todo caso será un número infinito *numerable*[5]. Por ejemplo, en un sistema binario la señal digital sólo puede tomar dos valores posibles (por ejemplo $+5$ y -5 voltios).

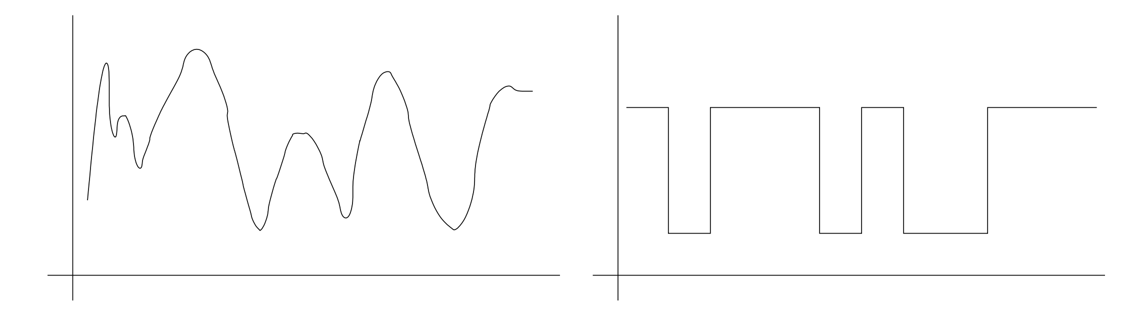


Figura 5.6. A la izquierda, una señal analógica, que puede tomar cualquier valor dentro de los límites establecidos. A la derecha, una señal digital, que sólo puede tomar, en un instante concreto, un valor dentro de un conjunto discreto establecido de antemano

En informática y telecomunicaciones son preferibles las señales digitales en la transmisión de información a causa de las muchas ventajas que conlleva su utilización. Ciertamente, los ordenadores son sistemas digitales binarios y, en ese sentido, trabajan internamente con señales digitales que sólo pueden estar, en un instante determinado, en uno de dos estados posibles. Por ello, parece claro que las redes de ordenadores deberían trabajar con señales digitales. Pero no solamente por eso, sino también por las ventajas que obtenemos con su utilización y que pueden condensarse en las siguientes:

- *Eficacia*: la circuitería digital está muy perfeccionada y es más económica que sus equivalentes analógicos. Es más fácil depurar el funcionamiento de un circuito digital que uno analógico y, además, existen una serie de componentes básicos muy eficaces que simplifican el diseño de nueva circuitería.

[5]Ser numerable o no significa, en matemáticas, que los elementos del conjunto pueden ponerse en correspondencia con los números naturales.

- *Fiabilidad*: cuando hay interferencias, es más fácil reconstruir la señal original. Las técnicas de interpolación y corrección de errores funcionan mejor con las señales digitales, ya que el conjunto de valores es conocido de antemano y es numerable, lo cual simplifica la aplicación de dichas técnicas.

- *Capacidad*: es posible *multiplexar* la señal en el tiempo o transmitir paquetes de datos sólo cuando es preciso, optimizando la cantidad de datos que podemos transmitir por el canal o medio físico (el denominado ancho de banda del canal o *bitrate*). Multiplexar significa, básicamente, emitir varias señales distintas (cada una portadora de una información distinta) dentro de una misma señal.

  Si bien la multiplexación en el dominio de la frecuencia es algo habitual y que se realiza también con las señales analógicas (por ejemplo, la radio y la televisión: recibimos ondas de radio, pero según la frecuencia elegida escuchamos una emisora u otra), no pasa lo mismo con la multiplexación en el tiempo, donde las diferentes señales se superponen unas a otras pero desplazadas en el tiempo (en diferentes "ventanas" temporales). Sin embargo, esto sí que se puede realizar fácilmente con señales digitales.

  Un ejemplo que puede servirnos para ilustrar, en este sentido, la diferencia con el empleo de señales digitales lo constituye la *Televisión Digital Terrestre* (TDT) si la comparamos con la televisión analógica convencional: con la señal de TDT viene más información (por ejemplo, sinopsis de la película que están emitiendo) en la misma frecuencia (la misma señal contiene diversas secuencias de bits).

5.3.2. Canales para transmisión de información

El canal es el soporte o medio físico por el que se transmite la señal. Podemos encontrar tanto canales analógicos como digitales, en función del tipo de señal que transmiten. Veamos a continuación algunos ejemplos de canales para ilustrar esta cuestión:

- El sistema de *telefonía convencional* utiliza el par de hilos de cobre, el cual es un canal analógico donde el voltaje que circula por el cable reproduce la forma de onda de la voz que se transmite.
- La *radio* convencional utiliza un canal analógico: la amplitud de las ondas hertzianas que circulan por el aire o bien sus variaciones de frecuencia reproducen la forma de onda de la voz transmitida.
- La *telefonía móvil* digital utiliza canales digitales pese a usar ondas de radiofrecuencia en las transmisiones. Tanto las formas de onda producidas por la voz como los mensajes cortos se transmiten mediante secuencias de bits.

Por un lado, cada canal presenta sus propias características. A nosotros nos interesa el *ancho de banda* o *bitrate*, que es la cantidad de información que el canal es capaz de transmitir por unidad de tiempo. Se suele expresar en (múltiplos de) *bits por segundo*, bps. para abreviar. También nos interesa conocer la *distancia* máxima a la que es posible transmitir la información fiablemente. Por otro, tenemos que, en función de la capacidad de transmisión, existen canales:

Símplex: donde la información siempre viaja en un sentido, es decir, donde siempre tenemos un emisor y un receptor. Un ejemplo sería el canal de comunicación entre ordenador e impresora: el ordenador da órdenes y transmite los datos a la impresora[6]

Semidúplex: donde los sistemas comunicados pueden adoptar tanto el papel de emisor como el de receptor, pero nunca simultáneamente (cuando uno actúa como emisor el otro lo hace como receptor y viceversa).

Dúplex: cuando ambos sistemas adoptan indistintamente los roles de emisor y receptor (incluso simultáneamente) y la información puede viajar por el canal en ambos sentidos al mismo tiempo.

Dicho esto, vamos a centrarnos en lo que sería propiamente la transmisión de señales analógicas y digitales a través de canales analógicos y digitales. La transmisión de señales digitales a través de canales digitales no plantea problema alguno, como tampoco lo plantea la transmisión de una señal analógica a través de un canal analógico. Desde nuestro punto de vista, las redes de ordenadores, nos interesa ocuparnos de las transmisiones digitales.

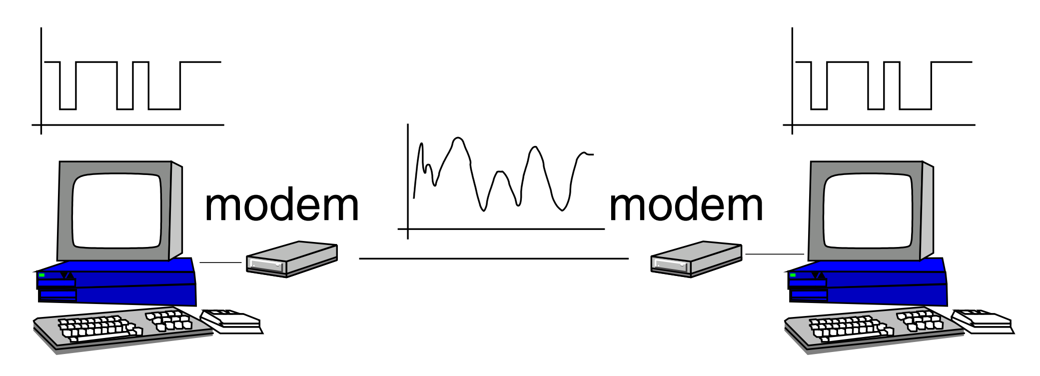


Figura 5.7. Los módems se encargan de transformar una señal analógica en digital y viceversa

En los últimos años, y especialmente en los próximos, se está procediendo a sustituir la mayor parte de las infraestructuras analógicas por digitales. Por ejemplo, en la actualidad prácticamente todas las centralitas y estaciones telefónicas son digitales, tanto para telefonía móvil como fija; las transmisiones de las señales televisivas y de radio, lo que se conoce en España como dominio público radioeléctrico, ya son duales (analógicas y digitales), con la obligación de ser solamente digitales en el año 2010 (el denominado "apagón analógico"); los sistemas de transmisión de datos son digitales; etcétera. Aun así, sigue siendo necesario, en algunos casos, utilizar canales analógicos para efectuar transmisiones digitales.

En estos casos, se hace precisa la utilización de sistemas adaptadores que conviertan la señal digital en señal analógica y viceversa. El ejemplo más común que podemos dar es el que se produce cuando se comunican dos ordenadores (que se transmiten información digital entre ellos) a través de la red telefónica convencional, la cual es un canal analógico.

[6]Si bien antiguamente esto era así, en la actualidad, esto ya no es tan claro, pues muchas impresoras envían datos al ordenador igualmente (por ejemplo, datos acerca del nivel de la tinta en los cartuchos).

Para poder llevar a cabo esta comunicación se requiere utilizar los denominados *módems* (módem es un acrónimo que proviene de la voz inglesa original para *modulator/demodulator*), que transforman la señal digital en analógica apta para su transmisión por la red telefónica y viceversa. La Figura 5.7 ilustra de forma grosera el funcionamiento de tales dispositivos.

El funcionamiento del módem consiste en transformar los datos que le llegan de manera digital en una serie de impulsos eléctricos que son transmitidos por la línea telefónica. Cuando estos impulsos llegan al módem del receptor (en la centralita telefónica) son transformados de nuevo en datos digitales. Estas dos transformaciones dan nombre al módem: *MOdulator/DEModulator*.

Nótese que, debido al funcionamiento "peculiar" del módem, la transmisión de datos se hace hacia un módem receptor que vuelve a digitalizar los datos. Por tanto, en una conexión a Internet mediante módem, nuestro módem *conecta con* ("llama a") otro módem. Dado que la cantidad de datos que se puede transmitir está bastante limitada por el propio funcionamiento de la línea telefónica, muchos módems realizan una compresión previa de los datos antes de enviarlos.

El origen de los módems

Cuando se plantearon las primeras redes de ordenadores y se estudiaba la infraestructura necesaria, se pensó que una de las maneras más sencillas de tener una red consistía en aprovechar otra ya existente.

Dado que la red telefónica está preparada para transmitir voz humana y no datos, se necesitaría algún tipo de adaptación: así surgieron los módems.

Con esta idea en mente se decidió aprovechar la red telefónica para conectar entre sí ordenadores, ya que era más barato diseñar y construir los módems que tender toda una nueva red con otra tecnología.

Ahora que hemos visto que es posible transmitir información digital por canales analógicos, en posteriores secciones nos centraremos en los procesos de comunicación digital propiamente dicha. En este sentido, se debe tener en cuenta que se puede efectuar una comunicación:

- en serie, donde la información se transmite bit a bit, o
- en paralelo, donde la información se transmite en grupos de n bits que circulan simultáneamente por varios cables.

5.3.3. Topologías de las redes informáticas

Además de establecer el medio físico con el que vamos a efectuar la conexión, es importante decidir sobre ciertas cuestiones al respecto de cómo se van a interconectar entre sí los ordenadores que conforman la red: esto es lo que se denomina comúnmente como la *topología* de la red.

Por ejemplo, ¿vamos a conectar todos los ordenadores entre sí dos a dos? Este esquema garantizaría la transmisión de datos en todo momento a pesar de las averías, pero requiere mucho cableado. Por el contrario, ¿vamos a conectar todos los ordenadores a un único ordenador central que va a ser el encargado de dirigir los datos de un sitio a otro? Este esquema requiere menos cableado, pero es más sensible a averías, ya que si se estropea el ordenador central la red deja de funcionar. Al hilo, otra posibilidad sería conectar todos los ordenadores a un único cable común, por lo que ya no se necesitaría de un ordenador central que se encargue de posibilitar y mantener las comunicaciones, pero entonces aparecerían otros problemas (colisiones, por ejemplo). De todo esto hablaremos en los siguientes apartados.

Topología en malla

Esta topología de red implica que cada ordenador se conecta a todos los demás. Por consiguiente, se debe tender cable entre cada par de ordenadores que deseamos conectar a la red. La Figura 5.8 permite ilustrar este esquema de conexión en red. Esta topología hace que la red sea muy robusta ya que, aunque uno o varios ordenadores se averíen ("caigan" en la jerga informática), la red como tal sigue funcionando pues es es posible establecer la comunicación entre cualesquiera dos ordenadores que estén operativos.

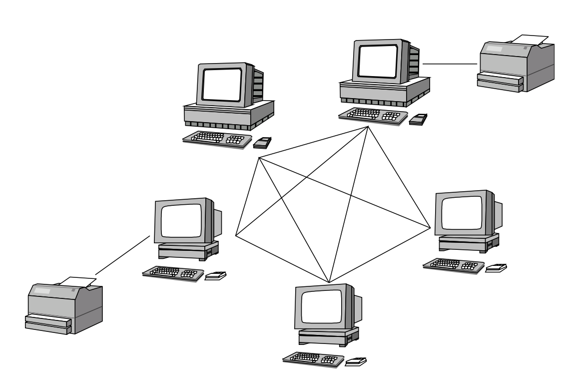


Figura 5.8. Topología de red en forma de malla

Otra ventaja, sin duda, es la rapidez de la red puesto que cuando dos ordenadores se comunican entre sí utilizan un cable en exclusiva (dedicado a la conexión). El tráfico que circula por el cable está formado únicamente por paquetes pertenecientes a la conexión establecida, lo cual implica que el ancho de banda no se comparte en absoluto.

Sin embargo, este tipo de red es excesivamente onerosa ya que tenemos que tender cable entre todo par de ordenadores conectados a la red. Si tenemos n ordenadores conectados, debemos establecer n^2 conexiones, con el consiguiente dispendio y sin poder evitar la redundancia de varias líneas.

La "redundancia", en este caso, quiere decir que podríamos eliminar estas líneas y aun así seguiría siendo posible la conexión entre todo par de ordenadores de la red (por ejemplo, llevando la "conversación" a través de vecinos e intermediarios).

Topología en anillo

Según este esquema de conexión en red, ilustrado en la Figura 5.9, cada ordenador que deseamos añadir a la red se conecta al "anterior" y al "siguiente" (por supuesto, los conceptos de "anterior" y "siguiente" pierden aquí su significado relativo al orden; simplemente nos estamos refiriendo a que el nuevo ordenador de la red se conecta con un único vecino en ambos sentidos). Cuando se establece una comunicación entre

dos ordenadores, la información parte del emisor y visita diferentes ordenadores en su ruta hacia el destino. Los ordenadores que no están directamente implicados en la conversación simplemente se limitan a "propagar" los datos hacia su vecino.

La mayor ventaja que tiene esta topología es económica: el cableado es sencillo y se reduce prácticamente al mínimo, por lo que es muy barata y cuesta poco de mantener. Sin embargo, el ancho de banda de las líneas puede llegar a compartirse entre varios ordenadores que están "conversando" simultáneamente. Además, la red es muy poco robusta, ya que el fallo de cualquier ordenador hace que ésta, como tal, deje de estar operativa (parcial o totalmente) en muchas ocasiones. Estos inconvenientes, en la práctica, han limitado enormemente los usos de esta topología.

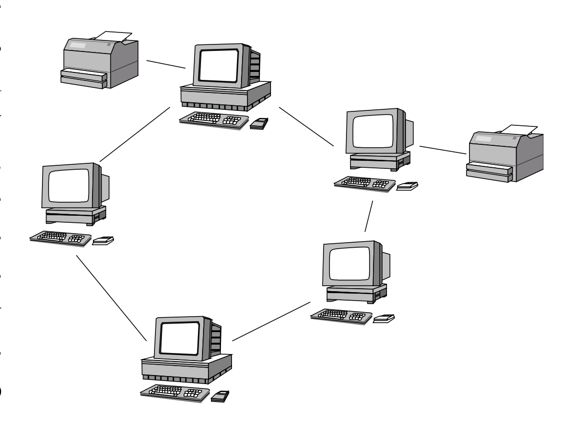


Figura 5.9. Topología de red en forma de anillo

Topología en *bus*

La topología en bus fue bastante utilizada desde los años setenta hasta comienzos de los noventa aproximadamente en las redes locales tipo Ethernet (empresas, centros de cálculo, departamentos de universidades...). Con este esquema, los equipos se conectan a un cable único (el "éter") cuyos extremos están cerrados con los denominados "terminadores" (los circulitos de la figura), tal como ilustra la Figura 5.10. Los terminadores son necesarios para delimitar los extremos de la red y poder así dirigir los paquetes de datos adecuadamente en uno u otro sentido. Pasemos a continuación a mencionar brevemente sus características más relevantes.

Con este esquema de conexión es posible que cualquier ordenador pueda entablar una comunicación con cualquier otro. La información se transmite por el cable que constituye la "espina dorsal" de la red y que hace el papel de *éter* (medio que "rodea" a todos los ordenadores de la red y a través del cual "viajan" los datos). Pero, para que los paquetes de datos lleguen a su destino correcto, es necesario poder identificar cada ordenador de la red con una "dirección". ¡Cuidado!, el lector no debe confundir esto con la dirección IP de los paquetes de datos, ya que esto último tiene que ver con cuestiones del protocolo TCP/IP empleado para establecer las comunicaciones: nosotros estamos haciendo referencia a una característica necesaria para el funcionamiento de la red con la topología de *bus*.

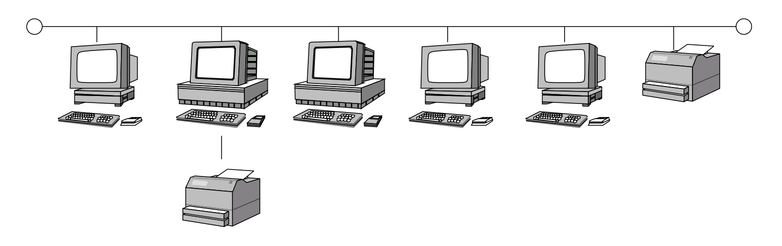


Figura 5.10. Topología de red en forma de bus. En los extremos del cable (éter) se conectan los terminadores que delimitan el "final" físico de la red

En realidad, lo que se identifica con una dirección es el adaptador empleado por los equipos que se conectan: son las denominadas tarjetas Ethernet (también denominadas tarjetas de red) que se emplean en las redes locales de este tipo. Las tarjetas Ethernet, al igual que la mayoría de adaptadores de red (NIC, acrónimo del inglés *Network Interface Card*), tienen un identificador (cuasi)único[7] asociado denominado *dirección MAC* (en inglés *MAC address*, siendo MAC el acrónimo de *Media Access Control*), dirección *hardware* Ethernet (EHA, por *Ethernet Hardware Address*), o bien dirección del adaptador, siendo la primera denominación la más comúnmente empleada.

La dirección MAC es un número que actúa como un nombre para un adaptador de red concreto. Así, por ejemplo, las tarjetas de red de dos ordenadores diferentes tendrán direcciones MAC ("nombres") distintas, como sería el caso de una tarjeta de red y una tarjeta inalámbrica en el mismo ordenador o como ocurriría con los múltiples adaptadores de red que tiene un *router*.

La topología en bus permite que las señales circulen en *ambos sentidos*, de ahí (entre otras cosas) la necesidad de utilizar terminadores. Sin embargo, el hecho de compartir un cableado único para efectuar las comunicaciones hace que puedan haber "colisiones". Esta situación se produce siempre que dos o más ordenadores deciden usar la red simultáneamente. El protocolo de comunicaciones (en algún "nivel") debe resolver y gestionar las colisiones.

El problema más importante que afecta a esta topología y que la ha hecho obsoleta en favor de la topología en estrella (la más usada actualmente), es que es poco fiable ya que cualquier fallo en el cable central aborta el funcionamiento de la red entera. Además, es una arquitectura poco segura puesto que cualquier ordenador tiene acceso a los mensajes que intercambian cualquier otro par de ordenadores. Esto hace que un usuario malicioso con acceso a la red pueda obtener información confidencial de otros usuarios que se transmite por la red.

En contrapartida, la instalación de este tipo de redes es sencilla y muy barata: basta con instalar el cableado correspondiente (la "espina dorsal") que pase por todos los lugares desde los que se necesita ofrecer conexión. No obstante, se ha de tener en cuenta que la distancia máxima a la que se puede ofrecer conexión con esta topología está limitada por condicionantes físicos a unos pocos centenares de metros. Es por eso que esta topología se usa(ba) en redes locales con pocos ordenadores.

Topología en estrella

La topología en estrella es la utilizada mayoritariamente en la actualidad en el terreno de las redes locales. El esquema de conexión obedece al que se muestra en la Figura 5.11, donde los ordenadores (en general, equipamiento informático) se conectan a un dispositivo especial *de bajo coste* que permite conmutar los circuitos para que sea posible el establecimiento de la comunicación entre cualquier par de ordenadores (de forma similar a como operaban las telefonistas en las centralitas telefónicas antiguas, tal como hemos tenido ocasión de ver en algunas de las películas de los años cuarenta o cincuenta).

[7] En la actualidad, es posible cambiar la dirección MAC en la mayoría de las tarjetas. Existe *software* que permite hacerlo, práctica que se conoce con el nombre de *MAC spoofing*.

Esta topología auna la práctica totalidad de las ventajas que presentan el resto y carece de sus desventajas. Es rápida como la topología en malla, ya que vía el elemento central se establece una línea casi dedicada en exclusiva a la comunicación entre el par de ordenadores que se conectan: tan sólo se comparte una parte de la línea si hay más de un ordenador estableciendo comunicación con un mismo receptor.

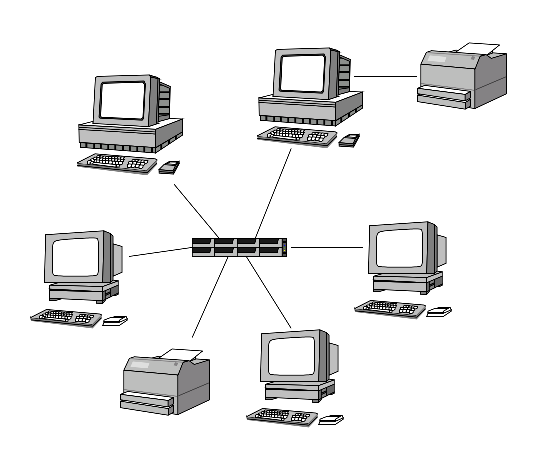


Figura 5.11. Topología de red en forma de estrella

Aunque a primera vista no lo parece, también es robusta, al igual que la topología en malla. No importa que se averíen uno o varios ordenadores, pues la red como tal seguirá funcionando igual. El punto débil es el elemento central de conexión común, obviamente: si se estropea éste, la red deja de funcionar. Entonces, ¿dónde está la cacareada robustez? se preguntará el lector. Nótese que el elemento central (en la práctica un concentrador o *hub* o bien un conmutador o *switch*) es un elemento de (relativamente) bajo coste. Por lo tanto, es fácil de sustituir en caso de avería, ya que normalmente se dispone de otro de repuesto. Al poder efectuar rápidamente la sustitución en la práctica, la red continúa funcionando (habitualmente) sin mayor problema.

Finalmente, cabe destacar que esta topología es barata, aunque no tanto como la topología en bus, ya que ésta únicamente requiere el cableado central que hace el papel de "espina dorsal". No obstante, puede considerarse de bajo coste, al igual que la topología en anillo, ya que sólo requiere "echar" cable entre cada ordenador que se desea conectar a la red y el elemento central de interconexión.

5.3.4. Taxonomía de las redes según su tamaño

Las redes de ordenadores son, en la práctica, de diferente tamaño y cobertura geográfica. Podemos encontrarnos desde una red local formada por un puñado de ordenadores separados por unas pocas decenas de metros hasta una red corporativa formada por cientos de ordenadores y distribuidos geográficamente por todo un país o continente (e incluso a escala mundial). Esto es posible ya que Internet realmente es una red de redes y, muchas de las redes que la conforman se componen igualmente de diversas subredes todas ellas agrupadas en una especie de "jerarquía". Para ser prácticos, podemos distinguir fundamentalmente tres tipos o categorías de redes según su tamaño, alcance y cobertura, las cuales se ilustran en la Figura 5.12:

- *Redes de área local* (o LAN, del inglés *local area network*): son redes que habitualmente cubren un edificio, o un campus, o las oficinas de una empresa, o bien una planta industrial... Son relativamente pequeñas, con velocidades rápidas y retrasos razonables (son redes de alta velocidad, por ejemplo de tipo *Fast Ethernet*).

- *Redes de área metropolitana* (o MAN, por *metropolitan area network*): son redes que cubren aproximadamente el área (o núcleo urbano) de una ciudad. Se usan, además de para comunicar ordenadores y transportar datos, para transmitir voz y vídeo (televisión por cable).

- *Redes de área extensa* (o WAN, del inglés *wide area network*): son redes que abarcan grandes extensiones de terreno. En muchas ocasiones son el resultado de la unión de diversas redes de área local (LAN). La WAN por excelencia es Internet; la red mundial de redes de ordenadores.

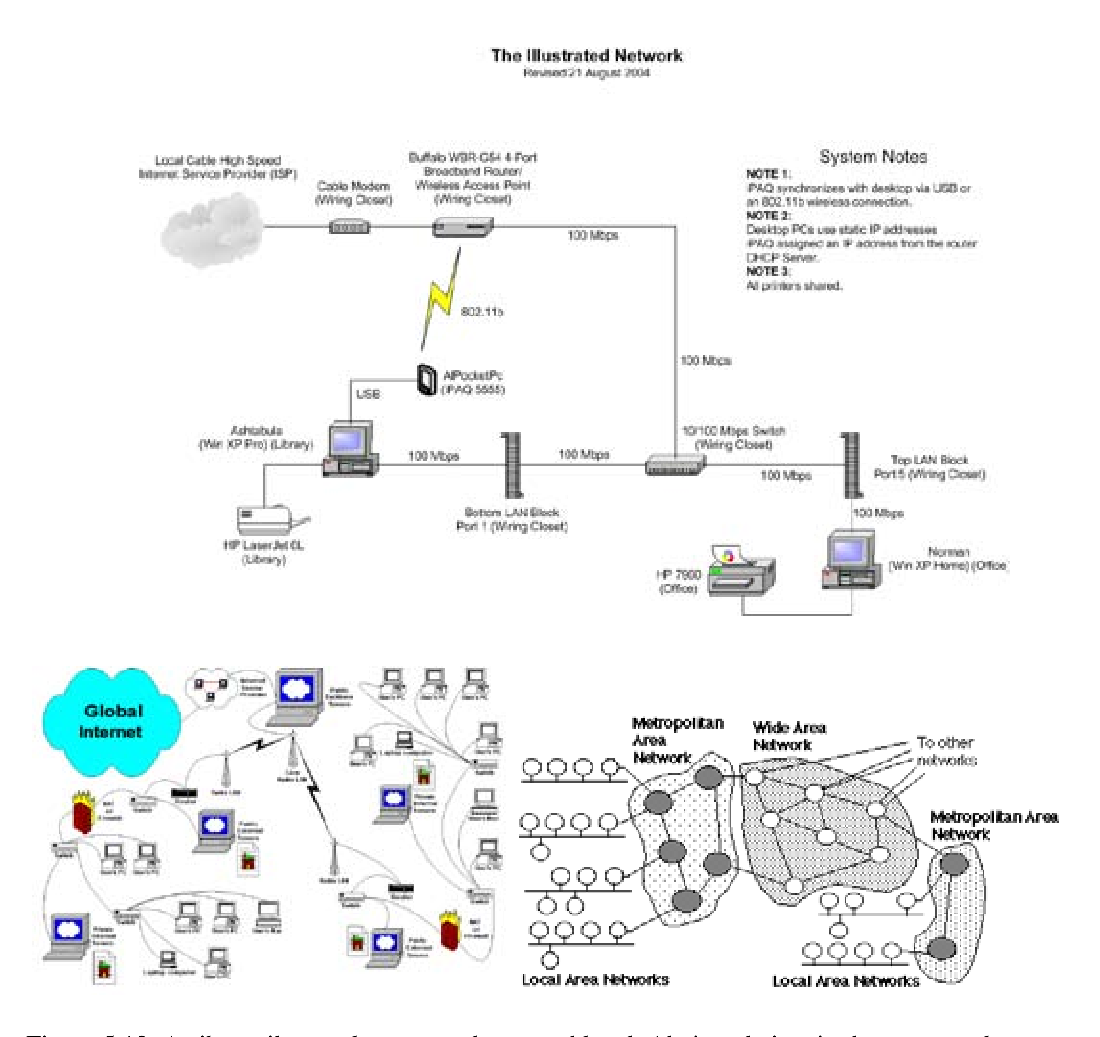


Figura 5.12. Arriba se ilustra el esquema de una red local. Abajo; a la izquierda tenemos el esquema de una red metropolitana que combina varios tipos de redes; a la derecha se ilustra el concepto de red de área extensa formada por la interconexión de varias subredes (metropolitanas y de área local)

5.4. COMPONENTES *HARDWARE* DE LAS REDES

Para que las redes puedan funcionar y operar correctamente, además del soporte físico (cableado, radiofrecuencia, etc.) correspondiente, se necesitan dispositivos y cir-

cuitería adicional que proporcionen, no sólo la conexión entre el ordenador y el medio físico, sino que también permitan la interconexión entre los ordenadores y entre las distintas redes que pueden comunicarse, el correcto encaminamiento de los paquetes de datos según marcan los protocolos de comunicación empleados, la gestión del "tráfico" de datos por la red e incluso la protección de los datos que se transmiten y de los propios ordenadores conectados a las diversas redes.

En consecuencia, en las redes de ordenadores podemos encontrar varios tipos de componentes, de entre los cuales podemos destacar principalmente:

- *Adaptadores de red*: son los dispositivos que permiten conectar de forma adecuada el ordenador al medio físico sobre el cual se va a establecer la comunicación. Estos aparatos dependen del soporte físico de la red. Por ejemplo, podría ser el caso de una tarjeta Ethernet (para la conexión a una toma RJ45 empleando el cable coaxial correspondiente), o una tarjeta Wi-fi (para la conexión a una red Wireless, radiofrecuencia), o bien un módem ADSL (para la conexión a la línea telefónica convencional y el envío de datos a través de la misma).
- *Los propios ordenadores*. Los ordenadores que se conectan a la red pueden interpretar diversos roles (y no necesariamente un rol único), de entre los cuales podemos destacar de forma grosera:
  - Servidores: son aquellos ordenadores que ofrecen servicios centralizados, como por ejemplo impresión remota, disco compartido, páginas web, transferencia de ficheros, correo electrónico, etc.
  - Estaciones de trabajo: son ordenadores de propósito general (para trabajos de cálculo, contabilidad, ofimática, gráficos y multimedia, etc.) que se conectan a la red para realizar trabajo en equipo o bien poder acceder a los diferentes servicios proporcionados por los servidores.
  - Ordenadores de red: se trata de ordenadores "especiales", sin disco duro o bien sin tener instalado el sistema en el mismo, que cargan el sistema operativo a través de un servidor de la red o de memoria ROM.
- *Concentradores o hubs*: son dispositivos de interconexión empleados en redes que permiten conectar entre sí los ordenadores y retransmiten los paquetes que reciben desde cualquiera de ellos a todos los demás. Hoy en día están dejando de ser utilizados debido al gran nivel de colisiones y tráfico de red que propician, siendo sustituidos por los *switches*.
- *Equipos de ampliación de red e interconexión de redes*. Se trata de dispositivos necesarios para el buen funcionamiento de las redes informáticas, ya que, básicamente, reparten y regulan el tráfico de paquetes por las mismas. Podemos encontrar, como elementos destacados, a:
  - los repetidores, que se encargan de amplificar y reconstruir la señal portadora de los datos,
  - los puentes, que permiten dividir y conectar el tráfico en redes diferentes,
  - los encaminadores o *routers*, que se encargan de dirigir y repartir el tráfico entre las diversas (sub)redes,

- las pasarelas, que permiten interconectar entre sí redes que, en principio, pueden tener arquitecturas muy diferentes o bien usar protocolos de comunicación distintos,
- los conmutadores o *switches*, que son dispositivos programables –y que, por tanto, pueden hacer tareas de cierta complejidad tales como filtrar determinados tipos de paquetes de datos, enmascarar puertos, etc.– encargados de conmutar líneas y repartir el tráfico entre los diversos equipos conectados a una red.

5.4.1. Soportes de red y adaptadores

En general, podemos afirmar que se requiere algún tipo de dispositivo periférico especializado para que un ordenador pueda comunicarse con otro. Dicho dispositivo va a depender del soporte o medio físico sobre el cual se vaya a transmitir la información en la red. Por ejemplo, no podemos usar el mismo dispositivo que se emplea para la conexión a una toma RJ45 en una red Ethernet, que el que emplearíamos para la conexión a una línea telefónica convencional que soporta ADSL. Veamos un resumen de algunos de los medios físicos que pueden utilizarse para transmitir información y sus correspondientes adaptadores:

- Para la red telefónica convencional (el par de cobre que tenemos en casa), también conocida por sus siglas (RTC) para abreviar, podemos emplear un módem convencional o bien un módem ADSL (lo habitual en la actualidad). ¡Ojo!, aunque ambos dispositivos son módems en el sentido del tratamiento de la señal, su funcionamiento es bien distinto y no son compatibles entre sí. Estos dispositivos se conectan a una toma telefónica convencional por un lado y, por el otro, al ordenador bien a través del puerto serie (módem convencional) o a través de un puerto USB (módem ADSL). Algunos módems ADSL son también enrutadores y pueden conectarse al PC a través de una tarjeta Ethernet.

- Si se trata de una red tipo Ethernet, entonces instalaremos en nuestro ordenador una tarjeta (de red) Ethernet. Hoy en día, la mayoría de los ordenadores ya vienen equipados con dicha tarjeta en la placa base.

- Si queremos conectarnos a una red de cable (existen varias compañías que operan con redes de este tipo en áreas urbanas), entonces tendremos que usar un dispositivo especial denominado cable-módem, el cual no podríamos considerar exactamente como un módem en el sentido clásico de la palabra, sino como una especie de enrutador-puente entre nuestro ordenador y la red de cable que proporciona el operador. Estos aparatos suelen ofrecer una interfaz de conexión tipo Ethernet, con lo cual podríamos conectarlos directamente a la tarjeta de red de nuestro ordenador empleando el cable adecuado, aunque también suelen venir equipados con un puerto USB para así poder conectarlos directamente a un puerto USB de nuestro PC.

- También existen otros medios o soportes alternativos que permiten la transmisión de señales portadoras de datos y, por consiguiente, el establecimiento de redes informáticas. El más popular de todos ellos en la actualidad es el *aire*: la

utilización de microondas y radiofrecuencia permite el funcionamiento de las denominadas *redes wireless*, popularmente conocidas como Wi-fi. Para poder conectarse a una red de este tipo se requiere el empleo de tarjetas inalámbricas. También existe la posibilidad de interconectar equipos a redes existentes mediante conexiones *vía satélite* o incluso a través del cable de la luz (tecnología PLC, *Power Line Communications*) empleando módems específicos; aunque estas tecnologías son más "exóticas" y están poco comercializadas.

Red Telefónica Convencional (RTC)

La red de telefonía convencional (RTC para abreviar) permite, en general, conectar dos ordenadores entre sí. Normalmente, uno de los ordenadores conectados hace el papel de puente o pasarela hacia la red de la compañía que proporciona la conexión telefónica (el proveedor de acceso a Internet) y que, a su vez, tiene salida (está conectada) a Internet.

Para poder emplear la línea telefónica convencional, originalmente ideada para la transmisión de voz y no de datos, como enlace de datos con otro ordenador se debe usar un adaptador que pueda conectarse a alguno de los puertos de conexión y que sea capaz de convertir la señal digital en analógica para su transmisión por la línea y a la inversa: el módem telefónico.

Los módems telefónicos son los dispositivos que permiten usar la RTC para transmisión de datos, ya que en realidad lo que ocurre es que el módem conectado a nuestro equipo marca un número al cual está atendiendo otro módem (el receptor) que, a su vez, está conectado a otro ordenador. Por tanto, la "conversación" se desarrolla, en realidad, entre ambos módems (tal como ocurre con los dispositivos de fax).



Figura 5.13. A la izquierda podemos ver la imagen de un módem telefónico externo. A la derecha tenemos una tarjeta PCI interna que desempeña la funcionalidad del módem telefónico

En la actualidad, esta modalidad de conexión está prácticamente obsoleta y apenas se usa. En las áreas urbanas se utilizan mayoritariamente la tecnología ADSL y la de las redes de cable y, en menor medida, *wireless*. En las zonas rurales se emplean fundamentalmente modalidades de ADSL (de baja velocidad por la lejanía a la centralita

telefónica) y, para accesos ocasionales o bien desde zonas donde no está disponible el ADSL, se emplean conexiones a la red de telefonía móvil UMTS (3G) mediante "tarjetas" 3G especiales (similares a las de los móviles) que se conectan al ordenador a través del puerto USB.

Podemos encontrar módems telefónicos (RTC) de dos clases: módems *externos*, que se conectan al ordenador a través del puerto serie, USB o PCMCIA –si se trata de ordenadores portátiles– como es el caso del que se muestra en la parte izquierda de la Figura 5.13, o bien módems *internos* (tarjetas o circuitos integrados), que se conectan al bus PCI de la placa base o vienen directamente integrados en la misma. La parte derecha de la Figura 5.13 muestra precisamente una tarjeta que se conecta a una ranura libre del bus PCI de la placa base de un PC para realizar las funciones de un módem.

El problema fundamental que plantea la comunicación de datos a través de la línea telefónica radica en que la señal portadora de los mismos debe pasar por un medio analógico que ha sido diseñado para otro uso: la transmisión analógica de voz. Y decimos analógico porque, aunque en la actualidad la conexión de centralita a centralita es digital, no ocurre lo mismo con la conexión del domicilio hasta la centralita, pues en lo que se denomina "bucle de abonado" sigue siendo analógica. Esto, que puede parecer un hecho baladí en comparación con el desarrollo tecnológico actual, en la práctica introduce ciertas limitaciones:

- En lo que concierne al ancho de banda o *bitrate*[8], sólo es posible alcanzar una velocidad máxima de 56 Kilobits por segundo (Kbps.), es decir, unos 56.000 bits (algo más de 7.000 bytes) por segundo aproximadamente.

  Es por ello que muchos módems comprimen y descomprimen "al vuelo" la información que se transmite o recibe para poder obtener, en la práctica, mayores velocidades de transmisión.

- En cuanto a la *fiabilidad* de la comunicación, se calcula que se comete *un error por cada* 100.000 *bits* transmitidos. Esto es mucho. Ofrecemos un dato para que el lector pueda comparar este rendimiento y sacar sus propias conclusiones: se considera que el rendimiento de una red Ethernet es bajo si se produce un error por cada *billón* de bits transmitidos.

  Para paliar en cierta medida este problema, los módems RTC incorporan circuitería para la detección y recuperación de errores en la transmisión. Es por ello que los módems telefónicos incorporan su propio procesador digital de señales.

Algunas características de las redes de teléfono

La red telefónica conecta cada casa con una estación local. Esta conexión es analógica.

continúa en la página siguiente ...

[8]Parámetro que mide la tasa de bits transmitidos por unidad de tiempo, es decir, la "velocidad" de la red.

. . . viene de la página anterior

Si la llamada es a otra casa conectada a la misma estación local, se conecta con ella. En otro caso, se busca una ruta hacia la estación local a la que está conectado el número marcado (si existe).

La conexión entre las distintas estaciones locales se hace de manera digital.

Debido a que la señal pasa por un canal analógico existen muchos problemas de ruido, pérdida de señal, etc.

Además, las propias compañías telefónicas filtran la señal, dejando pasar una banda de 3 KHz y añadiendo elementos como supresores de eco que introducen nuevos problemas.

Tal como hemos comentado antes, se calcula que hay un error en un bit de cada 10^5 bits transmitidos. Por comparación, en una red local tipo Ethernet se considera que el rendimiento es bajo si hay un bit erróneo cada 10^{12} ó 10^{13} bits transmitidos.

Si se utiliza el sistema operativo Linux y se pretende realizar la conexión a Internet vía RTC, se debe tener en cuenta que existe un tipo de módems internos[9] que carecen del procesador digital de la señal. Estos módems, popularmente conocidos como *WINmódems*, son más baratos, pero requieren de un *software* particular (*driver*) que se encarga de realizar las labores de procesamiento de la señal. Al tener que realizar estas tareas empleando *software* (programa) en lugar de *hardware* (procesador o circuitos especializados) estos dispositivos producen una sobrecarga innecesaria en la CPU del ordenador. Pero esto no es lo grave[10], ya que en muchas ocasiones ocurre que el *software* requerido para hacer funcionar el módem está disponible sólo para sistemas operativos de la "familia" Windows. Es por ello que algunos de estos dispositivos no funcionan en Linux. Se puede encontrar más información a este respecto en `http://www.linmodems.org`.

En resumen. . .

Un *WINmódem* es un módem "recortado" en el que parte de la funcionalidad que normalmente hace es ahora responsabilidad del microprocesador del ordenador (se efectúa por *software*).

Un módem USB (son WINmódems la mayoría) se conecta al puerto USB. Muchos módems USB no siguen las especificaciones estándar de la conexión USB.

Su funcionamiento es dependiente de un cierto programa (el controlador o *driver*) que proporciona el fabricante. Como sólo suelen proporcionar una versión para Windows, muchos son inutilizables en Linux.

[9]También los hay externos, pero son bastante menos frecuentes.
[10]Y más teniendo en cuenta la potencia de los procesadores actuales.

Asymmetric Digital Subscriber Line (ADSL)

Hasta mediados de los ochenta se pensaba que el cuello de botella para servicios de banda ancha (Internet, videoconferencia, etc.) era el *bucle de abonado*, esto es el par de cobre que une los domicilios de los usuarios con la central local. Por tanto, se creía que la red de telefonía imponía un límite severo (56 Kbps.) al ancho de banda para transmisión digital.

Pero, en realidad, había estudios teóricos que demostraban que la conexión convencional del teléfono a la centralita local (par de cobre) podía funcionar hasta 250 veces más "rápido" empleando la tecnología adecuada y, por tanto, permitía un ancho de banda mucho mayor.

A finales de los ochenta, los avances en microelectrónica hicieron posible el desarrollo de nuevos procesadores digitales de la señal capaces de aplicar nuevos algoritmos de proceso. Así aparecieron los módems ADSL (acrónimo del inglés *Asymmetric Digital Subscriber Line*) que explotan las propiedades teóricas del cableado telefónico para ofrecer, sobre el mismo, velocidades mucho mayores que el módem convencional.

Al contrario de lo que el nombre pueda indicar, la tecnología no hace referencia a la línea sino al dispositivo que permite la conexión. Por ello se dice que ADSL es una *tecnología de módem* que transforma el par de cobre del abonado en una línea de alta velocidad (la línea no cambia sino el módem, que es "especial".). Cuando pagamos por ADSL, no pagamos por la línea telefónica (es la que tenemos), sino por los módems que permiten establecer este tipo de conexión. En resumen, ADSL es una técnica de modulación para la transmisión de datos a gran velocidad sobre el par de cobre.

Hoy en día existen módems capaces de transmitir hasta 24 Mbps. aproximadamente en sentido descendente (desde la red hacia nuestro ordenador) y hasta casi 2 Mbps. en sentido ascendente (desde nuestro ordenador hacia la red). Los módems ADSL permiten, además, hablar por teléfono y enviar o recibir datos simultáneamente *usando el mismo cable telefónico que ya tenemos en nuestro domicilio*, ya que se usan *frecuencias distintas* para cada función. Básicamente, los módems ADSL difieren de los módems convencionales (RTC) en que los últimos sólo transmiten en la banda de frecuencias usada en telefonía (hasta 3 KHz aproximadamente), mientras que los módems ADSL operan en un margen de frecuencias mucho más amplio: 24 KHz-12 MHz.

Ésta es la razón por la que la tecnología ADSL puede coexistir en un *mismo bucle de abonado* (una misma línea) con el servicio telefónico convencional ya que no existe solapamiento en la banda de frecuencias en la que ambos operan. Esto quiere decir que podemos estar conectados a Internet y efectuar o recibir llamadas al mismo tiempo. Puesto que ADSL utiliza el mismo tipo de cable que tenemos habitualmente en casa o en las empresas, no hay necesidad de montar un cableado nuevo, lo cual constituye otra ventaja sobre la red telefónica convencional.

Para efectuar las comunicaciones, ya sea de voz, ya sea de datos, ADSL crea, virtualmente, tres canales *independientes* sobre la misma línea telefónica:

- El primero es el *canal estándar*, que se utiliza para realizar la comunicación normal (voz).
- El segundo es el *canal de alta velocidad*, que va desde 1 hasta 8 Mbps. y que se utiliza para *recibir* información (datos).

- El tercero es el *canal de velocidad media*, que va desde 16 Kbps. hasta 1 Mbps. y que se utiliza para *enviar* información (datos).

El lector puede apreciar que el canal de recepción y el de envío ofrecen velocidades dispares, de ahí el calificativo *asymmetric* (asimétrica). Esta asimetría no es, generalmente, un inconveniente, ya que el usuario doméstico suele recibir (mucha) más información que la que envía. En otras palabras, ADSL se basa en que cuando usamos Internet solemos recibir más información que la que enviamos. Por ejemplo, si queremos descargar un fichero de 2 MB, en realidad sólo enviamos la petición (unos pocos kilobytes), pero recibimos todo el fichero (algo más de 2 MB teniendo en cuenta la información de control de la transferencia). Normalmente, el ancho de banda proporcionado por ADSL está disponible de forma permanente.

Más cosas sobre ADSL

Al tratarse de una modulación en la que se transmiten diferentes caudales en los sentidos "usuario-red" y "red-usuario", el módem ADSL situado en el extremo del usuario es distinto del ubicado en la central local de la compañía de telecomunicaciones que proporciona el acceso a Internet.

Delante de cada uno de ellos se ha de colocar un dispositivo denominado *splitter*. Este dispositivo no es más que un conjunto de dos filtros: uno pasa alto y otro pasa bajo. La finalidad de estos filtros es la de separar las señales transmitidas por la línea: baja frecuencia para telefonía (voz) y alta frecuencia para ADSL (datos).

Por tanto, la tecnología ADSL realiza una división de la banda de frecuencias a la que es posible operar sobre el cable de línea telefónica, de forma que no impide tener una conversación con un amigo y a la vez estar conectado a Internet y buscar cosas.

Sin embargo, ADSL presenta ciertas limitaciones que hacen que, en algunas ocasiones, no sea una tecnología práctica para establecer la conexión a Internet. Estas limitaciones tienen que ver, fundamentalmente, con problemas de atenuación de la señal en el cable y con el ruido y distorsión excesivos que pueden ocurrir en ciertas situaciones.

En un par de cobre (cableado telefónico estándar), la atenuación por unidad de longitud aumenta a medida que se incrementa la frecuencia de las señales transmitidas. Y cuanto mayor es la longitud de la línea, tanto mayor es la atenuación total que sufren estas señales. Nótese que, precisamente, la tecnología ADSL utiliza altas frecuencias para el envío y recepción de datos. Ambas cosas explican que el caudal máximo que se puede conseguir mediante los módems ADSL varíe en función de la longitud del bucle de abonado hasta la centralita del proveedor telefónico, es decir, que el ancho de banda *real* o efectivo (la velocidad de transmisión en suma) se ve afectado por la distancia a la central local.

En presencia de ruido (en el peor de los casos) y hasta una distancia de 2,6 kilómetros de la central local, que es la situación típica que podemos encontrar en zonas urbanas, se obtiene un ancho de banda de 2 Mbps. en sentido descendente (recepción

de datos) y 0,9 Mbps. en sentido ascendente (envío de datos). Esto hace de ADSL una tecnología eminentemente *urbana*, ya que ésta es la longitud media del bucle de abonado en las ciudades[11].

En la práctica, la mayor parte de usuarios de ADSL están en condiciones de recibir un caudal entre 2 y 6 Mbps. Si la distancia a la centralita es grande (mayor de 3,5 km.) la velocidad será mucho menor o incluso no se podrá instalar ADSL en el domicilio o en la empresa, ya que no es viable su utilización. Y aun a pesar de que las condiciones anteriores se cumplan, quizás no se pueda instalar ADSL debido a un exceso de interferencias en la línea telefónica. Será el propio instalador el que, usando aparatos especiales, nos dirá si podemos o no instalar ADSL. Ciertamente, es muy raro que estos casos lleguen realmente a producirse en zonas urbanas.

El lector atento se preguntará cómo es posible que, existiendo un límite teórico de 8 Mbps. al ancho de banda que podemos conseguir con ADSL, se publiciten ofertas de ADSL de hasta 20 Mbps. por parte de algunas empresas de telecomunicaciones que operan en nuestro país. Bien, el hecho cierto es que, desde hace unos pocos años, la tecnología ADSL evolucionó[12] hasta desarrollar las "extensiones" (permítasenos emplear este término) denominadas *ADSL2* y *ADSL2+*.

Estas tecnologías mejoradas de ADSL de reciente aparición tienen la capacidad de ofrecer televisión y vídeo de alta calidad a través del cable telefónico (debido al importante aumento del ancho de banda que proporcionan, entre 3 y 4 veces más en la práctica) junto con otras mejoras tecnológicas, tales como corrección de errores, supervisión del estado de la conexión, mejora de velocidad con múltiples líneas telefónicas, etc. Desde nuestro punto de vista, centrado fundamentalmente en el ancho de banda proporcionado, podemos resumir diciendo que:

- La tecnología ADSL2 proporciona, en teoría, un ancho de banda máximo en el canal de envío de datos entre 1 y 2 Mbps. (dependiendo de la versión implementada), y un ancho de banda máximo en el canal de recepción de datos de 12 Mbps. Debido al hecho del rápido desarrollo del ADSL2+ (pronúnciese "ADSL2 plus") a continuación, prácticamente no ha sido comercializada en nuestro país. Los operadores que ofertan ADSL de alta velocidad están realmente ofreciendo ADSL2+.

- La tecnología ADSL2+ proporciona en teoría un ancho de banda máximo en el canal de envío de datos entre 1 y 2 Mbps. (idéntico al ADSL2), y un ancho de banda máximo en el canal de recepción de datos de 24 Mbps. (el doble). Es el ADSL de alta velocidad que están ofreciendo actualmente las "telecos" (empresas de telecomunicaciones).

ADSL2 (y, en consecuencia, ADSL2+) es capaz de dar cobertura a bucles de abonado más largos que los que permite el ADSL convencional. Ello también implica que ADSL2 proporcione mayores velocidades a puntos alejados con respecto al ADSL

[11]No obstante, en España se realizó recientemente una campaña denominada "ADSL rural", a cargo del operador dominante, con el objetivo de establecer más centralitas en los pueblos de cierta relevancia.

[12]La teoría evolutiva de Darwin no deja de comprobarse experimentalmente una y otra vez, mal que les pese a los fanáticos religiosos que hay detrás de "cosas" como el Creacionismo o el Diseño Inteligente (su sucesor "enmascarado").

convencional (es menos sensible a la distancia a la centralita). No obstante, cabe hacer notar que el ruido afecta de manera más notoria a ADSL2+, ya que éste utiliza la parte más alta del espectro de frecuencias. Así que, realmente, ADSL2+ tan sólo supone una mejora en el ancho de banda si consideramos una distancia de hasta 3 km. a la central local. A partir de ahí las diferencias con ADSL o ADSL2 son mínimas. También merece la pena resaltar que, a diferencia de la migración a ADSL2, ADSL2+ requiere pequeños cambios en la estructura de la red.

En la actualidad, la mayoría de los operadores que trabajan en zonas urbanas proporcionan el servicio de ADSL2+ (al que algunos llaman ADSL de alta velocidad). Merece la pena resaltar que estos nuevos desarrollos de la tecnología ADSL la sitúan en posición de clara competencia con la tecnología empleada en las redes de cable.

Para poder conectar nuestro ordenador a Internet a través de una conexión telefónica empleando ADSL necesitamos utilizar el dispositivo adaptador correspondiente a este medio. En este caso, se trata de un *módem ADSL* (que no tiene nada que ver, repetimos, con un módem RTC, el cual no es compatible con ADSL). En la actualidad, podemos encontrarnos con tres opciones a la hora de escoger un módem ADSL, siendo la más frecuente la segunda (es la opción que suelen "vender" las empresas que proporcionan servicios de Internet a través de ADSL):

- La primera consiste en usar un módem ADSL *interno*. En este caso, el dispositivo se conectará habitualmente a nuestro ordenador usando una ranura del bus PCI.
- La segunda consiste en usar un módem ADSL externo. En este caso, el módem puede conectarse a nuestro ordenador a través del puerto USB.
- La tercera consiste en una combinación de tarjeta de red (interfaz de conexión) y módem ADSL (con posibilidad de conexión a tarjeta Ethernet).

  Para ello, el módem ADSL dispone de una conexión RJ45 a la que podemos conectar un cable coaxial que se conecta por el otro extremo a la tarjeta de red de nuestro ordenador. Aparte, el módem dispone de la conexión habitual con la red telefónica. En este caso, el módem ADSL actúa además como un *router*, ya que conmuta paquetes entre las dos redes (la Ethernet entre el PC y el módem y la red ADSL entre el módem y la red telefónica). Es por eso que, muchas veces, se denomina *router ADSL* a estos dispositivos.

Configuración desde Windows y Linux

Para la configuración de la conexión desde Windows, además de los *drivers* (programas manejadores) de los dispositivos implicados, la empresa que nos proporciona el servicio de ADSL suele realizar la configuración personalmente o bien proporcionar los medios para ello (normalmente mediante un *kit* de instalación).

continúa en la página siguiente . . .

. . . viene de la página anterior

En el caso de Linux, la opción que va a funcionar en la mayoría de los casos consiste en tener un módem ADSL externo (que funciona como *router*) y que se conecta a nuestro PC mediante cable coaxial que comunica la tarjeta ethernet del mismo con un puerto RJ45 del módem.

En este caso, basta con que Linux soporte la tarjeta Ethernet de nuestro ordenador. La gran mayoría de ellas ya vienen soportadas de forma nativa en el *kernel* de Linux y éste la detectará automáticamente al arrancar.

Después habrá que configurar la conexión ADSL propiamente dicha. Para ello, habrá que preguntarle los datos de configuración al proveedor de Internet (PI), aunque lo más habitual es que el PI haya establecido mecanismos que permiten configurar la conexión de forma (semi)automática.

Existen dos posibilidades para ello: DHCP (*Dynamic Host Configuration Protocol*) o PPPoE (*Point-to-Point Protocol over Ethernet*). El PI (su servicio técnico) debe indicarnos cuál de ambos utiliza.

En el primer caso, necesitamos configurar y usar adecuadamente un programa *cliente* para el servicio DHCP (lo abordamos en la sección 5.5.3). En las diferentes distribuciones de Linux, basta con instalar algún paquete denominado `dhcpcd`, `dhcp-client`, `dhcp-utils`, `pump` o similar.

En el segundo caso, necesitamos un programa cliente PPPoE. En las diferentes distribuciones, hemos de instalar el paquete llamado `pppoe`, `rp-pppoe` o similar. También será útil instalar el paquete denominado `pppoeconf` o similar (de hecho, al instalar este paquete también se instalan todos los necesarios, incluyendo el cliente PPPoE). Éste es un programa "amigable" que facilita la configuración de la conexión ADSL mediante PPPoE. En cualquier caso, antes de "montar" ADSL en Linux, se recomienda la lectura del DSL-HOWTO: `http://tldp.org/HOWTO/DSL-HOWTO/index.html`

Si adquirimos un módem PCI interno hay que fijarse en la marca y el modelo, y comprobar si existe el correspondiente *driver* para Linux. Si nuestro sistema Linux no lo detecta, o bien lo detecta pero no consigue configurar la conexión a Internet, lo más seguro es que no esté soportado. Podemos buscar información por Internet, pero si no encontramos el *driver* y sus instrucciones de instalación, no podremos hacerlo funcionar en Linux. Igualmente, habrá que ver si necesitamos DHCP o PPPoE para configurar la conexión.

Si tenemos un módem ADSL externo, conectado a nuestro PC a través de un puerto USB, estamos en la misma tesitura que con el módem interno. Necesitamos instalar el *driver* correspondiente en Linux (hay que fijarse en la marca y modelo del módem) y configurar posteriormente la conexión ADSL, bien manualmente o bien mediante DHCP o PPPoE. Si el módem viene soportado de forma nativa en el *kernel* de Linux, normalmente el *driver* se cargará automáticamente al conectarlo al puerto USB y encenderlo.

Very high speed Digital Subscriber Line (VDSL)

La tecnología VDSL (acrónimo del inglés *Very high speed Digital Subscriber Line*), al igual que ADSL, utiliza el cableado telefónico convencional que llega a nuestros domicilios (mediante la transmisión de impulsos sobre pares de cobre) para proporcionar acceso a Internet de banda ancha. Se trata, pues, de una evolución del ADSL, que puede suministrarse tanto de manera asimétrica, con velocidades de hasta 52 Mbps. en el canal de descarga (recepción) de datos y de hasta 12 Mbps. en el canal de subida (envío) de datos, como de manera simétrica, con velocidades de hasta 26 Mbps. tanto de subida como de bajada. Tal y como ocurre con la tecnología ADSL, con la que está "emparentada", las velocidades máximas sólo podrían obtenerse en condiciones ideales, esto es, sin resistencia a la señal en el par de cobre, poco ruido y con una distancia nula a la central, con lo que, en la práctica, se obtienen velocidades algo menores.

La tecnología VDSL utiliza, en realidad, 4 canales (bandas de frecuencia) diferentes para la transmisión de datos, dos para recepción (del proveedor hacia el cliente) y otros dos para el envío (del cliente hacia el proveedor), aunque en la práctica cada par funciona como si fuese sólo uno. La utilización de más canales que ADSL permite aumentar la potencia de transmisión de manera sustancial. La técnica estándar de modulación puede ser QAM/CAP (*Carrierless Amplitude/Phase*) o DMT (*Discrete MultiTone modulation*), las cuales no son compatibles, pero tienen un rendimiento similar. Actualmente la más usada es DMT.

En la actualidad, la tecnología VDSL se está empleando fundamentalmente para transmitir señales de televisión de alta definición por la red[13], es decir, vídeo comprimido. Sin embargo, el vídeo comprimido es una señal en tiempo real poco apta para su transmisión por la línea telefónica, aun estando preparada para comunicaciones de datos, debido al alto número de errores que se producen. Para lograr tasas de error compatibles con el vídeo comprimido, VDSL incorpora la técnica denominada *Forward Error Correction* (FEC) con el objetivo de corregir todos los errores producidos por la aparición de pulsos ruidosos en la línea.

En general, la tecnología VDSL padece una mayor atenuación que ADSL por la distancia a la central local, ya que es más sensible a los problemas de resistencia a la señal en la línea telefónica. Podemos decir que, actualmente, esta tecnología se está implantando en varios países europeos, aunque es en un país tan desarrollado tecnológicamente como Japón donde está muy extendida (líneas de alta velocidad para la conexión a Internet). En España, la multinacional Telefónica está haciendo pruebas con esta tecnología con el objetivo, sobre todo, de potenciar su servicio de televisión por Internet (Imagenio). Se dice que, en breve, Telefónica ofrecerá Internet de banda ancha y televisión en los hogares usando VDSL, pero en el momento de redactar estos apuntes los autores no tenemos constancia de ello (tan sólo rumores).

Recientemente, la tecnología VDSL ha evolucionado hacia VDSL2[14], el estándar de comunicaciones DSL más reciente y avanzado. Está diseñado para soportar servicios de voz, vídeo, datos, televisión de alta definición (HDTV, por *High Definition*

[13] Al fin y al cabo, las imágenes de televisión no dejan de ser datos.

[14] De hecho, cuando actualmente se habla de la tecnología VDSL, en realidad, se está haciendo referencia a VDSL2.

TeleVision) y juegos interactivos sobre la línea telefónica convencional. Muchos proveedores telefónicos en España ya comercializaban ofertas de vídeo, voz y datos en una misma línea y, por tanto, VDSL2 permitirá a estos operadores actualizar gradualmente las líneas ADSL existentes, sin un coste muy elevado.

VDSL2 permite igualmente la transmisión simétrica o asimétrica de datos, llegando a anchos de banda superiores a los 200 Mbps. Obviamente, este ancho de banda depende de la distancia a la central. Así, los 250 Mbps. que se alcanzan a "distancia cero" de la central se reducen a un máximo en torno a los 100 Mbps. a 500 metros y a unos 50 Mbps. si el domicilio está a 1 km. Después, el descenso de velocidad es menos abrupto y, de hecho, presenta un porcentaje menor de pérdida en comparación con el VDSL original: a 1,6 km. el rendimiento es similar al máximo teórico que podría proporcionar ADSL2+, y si nos situásemos a 4 ó 5 km. de distancia el ancho de banda (para el canal de recepción de datos) estaría entre 1 y 4 Mbps.

De este modo, la tecnología VDSL2 no está meramente limitada a cortos bucles, sino que puede ser utilizada con calidad en medias distancias. En cuanto a la relación de simetría entre los canales de recepción y envío, podemos decir que con VDSL2, a medida que la longitud del bucle de abonado se acorta, aumenta la relación de simetría, llegando a ofrecer velocidades por encima de los 100 Mbps. tanto en descarga como en envío (supuesto que se den las condiciones idóneas de ruido y distancia a la central).

Fiber To The Home (FTTH)

La tecnología de telecomunicaciones FTTH (acrónimo del inglés *Fiber To The Home*), también conocida como fibra hasta el hogar, se basa en la utilización de *cables de fibra óptica* y *sistemas ópticos de distribución* adaptados a esta tecnología para la distribución de servicios de telefonía, Internet de banda ancha y televisión a los hogares de los abonados.

La implantación de esta tecnología está tomando fuerza, especialmente en países como Estados Unidos y Japón, donde muchos operadores reducen la promoción de servicios ADSL en beneficio de la fibra óptica con el objetivo de proponer servicios muy atractivos de banda ancha para el usuario (cine, música, televisión, etc.).

Desde el punto de vista técnico, la tecnología FTTH propone la utilización de fibra óptica empleando una multiplexación por división de longitud de onda. La interconexión entre el hogar del abonado y el nodo de distribución puede realizarse bien mediante una *conexión punto a punto* empleando la conocida topología en estrella (más cara), o bien empleando una red óptica pasiva (PON, del inglés *Passive Optical Network*) que reparte la información entre varios usuarios (más barata).

Ésta última es la opción menos costosa ya que supone emplear un dispositivo óptico divisor (*splitter*) del haz de luz entrante para su distribución hacia múltiples fibras. Por tanto, se basa en compartir los costes del segmento óptico entre los diferentes terminales de los abonados, de forma que se pueda reducir el número de fibras ópticas. Así, por ejemplo, una señal de vídeo se puede transmitir desde una fuente a múltiples usuarios. Las principales ventajas de esta solución son la fiabilidad que ofrece y su facilidad de despliegue.

Por otro lado, la topología en estrella proporciona 1 ó 2 fibras dedicadas a un mismo usuario, resultando una solución más cara. Su ventaja, no obstante, radica en que se

puede proporcionar un mayor ancho de banda para cada abonado, ya que sus datos no se comparten con los de otros usuarios de la misma terminal.

Actualmente, no podemos dar datos fiables acerca del ancho de banda que se podría alcanzar con esta tecnología, ya que requiere despliegue de cable de fibra óptica y el acondicionamiento de la red existente por parte de los proveedores de Internet, telefonía y televisión (las "telecos"). Por tanto, el ancho de banda que puede ofrecerse puede variar en función de los medios que aporte cada operador y de la forma de comercialización escogida. En España, Telefónica ha realizado algunas pruebas preliminares alcanzándose anchos de banda simétricos entre 50 y 100 Mbps, pero no se sabe nada más acerca de los planes de futuro en el despliegue de la red de fibra óptica de la compañía, por lo que en el momento de redactar estos apuntes poco más podemos decir. El operador de cable Ono también ha realizado pruebas a 100 Mbps. empleando la tecnología denominada DOCSIS, que puede considerarse un híbrido entre cable coaxial y fibra óptica.

FTTH está "emparentado" con otras tecnologías similares conocidas como *Fiber to the x*, siendo "x", *Node* (centralita), *Curb* (calle, acera, manzana) o *Building* (edificio). La diferencia fundamental entre todas éstas radica en el tipo de cableado que se utiliza para dar el servicio en último término al hogar del abonado y en la distancia hasta la centralita que conecta con el cableado de fibra óptica del proveedor. Así tenemos tecnologías como DOCSIS (del inglés *Data Over Cable Service Interface Specification*), mencionada en el párrafo anterior, o VDSL (véase la sección anterior) que emplean, respectivamente, cable coaxial o el par de cobre tradicional de la línea telefónica para llegar en último término al domicilio del cliente.

Podríamos decir, pues, que estas tecnologías serían del tipo *Fiber to the Node*, aunque, en función de la distancia hasta el domicilio a la que se haga llegar el cableado de fibra óptica, podrían catalogarse también como *Fiber to the Curb*[15] o *Fiber to the Building*, dependiendo de si el proveedor (la compañía de telecomunicaciones) lleva hasta nuestra calle o hasta nuestro edificio el cable de fibra óptica.

Ethernet

Una de las opciones más populares, incluso en la actualidad, para crear una red de área local es utilizar la tecnología (mecanismo estándar de conexión entre ordenadores, en este caso) *Ethernet*. Originalmente desarrollada por Robert Metcalfe (tal como vimos en la sección 5.2), la idea básica tras el diseño de Ethernet es tener un cable –que hace las veces del "éter" y de ahí su nombre, tal como comentamos en la sección 5.3.3– que todos los ordenadores conectados utilizan para comunicarse. Este tipo de redes fueron muy populares sobre todo en los años ochenta y noventa y, con el cambio de siglo, están siendo progresivamente "desplazadas" por las redes Wireless[16]. El "boom" de las redes Ethernet se produjo en gran medida gracias a la utilización del esquema de conexión entre ordenadores conocido como *topología en bus*, como estudiamos en la sección 5.3.3.

[15]En la mayoría de las ocasiones no se suele hacer distinciones entre *Curb* y *Node* ya que la diferencia es muy sutil en la práctica y apenas se aprecia.

[16]Aunque las redes Wireless utilizan tarjetas inalámbricas y un protocolo de comunicaciones que heredan bastantes de las características de Ethernet.





Figura 5.14. A la izquierda la imagen de una tarjeta Ethernet para PC. A la derecha tenemos una tarjeta Ethernet específica para su conexión al puerto PCMCIA que traen la mayoría de los portátiles

Para poder conectar un ordenador a una red tipo Ethernet, se requiere la utilización de un adaptador determinado (como no podía ser de otra manera): la denominada *tarjeta Ethernet*, popularmente conocida como *tarjeta de red*. En la Figura 5.14 podemos ver imágenes de tarjetas Ethernet.

Las tarjetas de red para PC pueden venir integradas en la placa base (lo habitual en la actualidad) o bien se conectan en un bus concreto del ordenador (habitualmente PCI). En el caso de los portátiles pueden venir integradas (lo más normal) o ser dispositivos externos que se conectan a un puerto PCMCIA específico para ordenadores portátiles.

Una vez conectada (o "integrada") al ordenador, la tarjeta muestra uno o dos conectores (siempre RJ45, adicionalmente puede venir una conexión tipo BNC) a uno de los cuales (RJ45, habitualmente) le conectaremos un cable (lo más normal es que sea de par trenzado) que, a su vez, estará conectado (directa o indirectamente a través de algún tipo de conexión) al medio de transmisión que empleemos.

Este medio de transmisión será habitualmente un *concentrador* (popularmente conocido como *hub*) o un *conmutador* (*switch*), ya que en la actualidad el esquema de conexión más usado es la denominada *topología en estrella* (véase la sección 5.3.3). Sin embargo, queremos recordar que, tal como vimos en la sección 5.4.1, las tarjetas Ethernet también pueden usarse para la conexión a un módem ADSL, por ejemplo (y, como veremos más adelante, también a un cable-módem).

Tipos de cable en una red Ethernet

Para efectuar la conexión desde la tarjeta de red de nuestro PC al medio de transmisión (o a la "toma" correspondiente) en una red Ethernet se pueden utilizar distintos tipos de cable. Destaquemos algunos:

continúa en la página siguiente . . .

. . . viene de la página anterior

• Cable 10Base5: se trata de un tipo de cable coaxial grueso. El cable coaxial ha sido tradicionalmente el más utilizado hasta la fecha (donde ha tomado su relevo el par trenzado). Este tipo de cable es bueno para distancias medias y se utilizaba con frecuencia en las redes con topología en bus.

• Cable 10Base2: es un tipo de cable coaxial fino. Constituye la solución más económica para realizar el cableado de una red Ethernet. Se ha utilizado mucho en los hogares y en pequeñas y medianas empresas para montar redes locales con pocos puestos empleando el esquema de topología en bus. Tras la llegada del *wireless* (no se requiere cableado para montar la red local) su uso se ha visto muy disminuido.

• Cable 10Base-T: éste es un cable de par trenzado, el cual es muy fácil de mantener. Es el más utilizado actualmente (en la variedad 100Base-TX que permite su conexión en redes Ethernet funcionando a 100 Mbps.) en las redes Ethernet, empleando conectores RJ45 para su conexión con los conmutadores o *switches* que se encargan de repartir el tráfico en la red mediante una topología en estrella.

• Cable 10Base-F: se trata de cable de fibra óptica, el cual se emplea habitualmente cuando se trata de conectar equipos a mayores distancias o bien de interconectar redes entre sí separadas a una cierta distancia (2 km. máximo).

Con cualquiera de estos sistemas de cableado se obtienen velocidades de hasta 10 Mbps.

Hay que tener en cuenta que con los cables tipo 10Base-T y 10Base-F cambia la forma de conexión. En este caso se utiliza la topología en estrella mediante la conexión a un elemento central (*hub* o *switch*) al que se conectan todos los ordenadores.

Desde el año 1995 existe otro estándar para Ethernet, que se ha dado en llamar *fast Ethernet* y que permite conectar los ordenadores a 100 Mbps. Este estándar necesita cable "especial" de par trenzado (100Base-TX) o fibra óptica (100Base-FX) y tenía la gran ventaja de que era compatible con los equipos existentes a finales de los noventa, ya que había gran cantidad de *switches* que podían funcionar simultáneamente con redes a 10 y 100 Mbps.

Recientemente (año 2001) ha aparecido un nuevo estándar de conexión, denominado *Ethernet 1GB* que permite velocidades de conexión de hasta 1.000 Mbps. Para emplearlo se requiere la utilización de *switches* (no valen *hubs*) en una topología en estrella y cable "especial" de par trenzado (1000Base-T) o fibra óptica (1000Base-SX o 1000Base-LX). El cable de fibra óptica 1000Base-LX permite establecer conexiones a una distancia máxima de 5 km.

Tradicionalmente, las redes Ethernet han operado con dos velocidades de conexión: 10 Mbps. y 100 Mbps. Éste último estándar es lo que se conoce como *Fast Ethernet*. Como se puede apreciar, las redes Ethernet son mucho más rápidas que la red telefónica convencional (los módems tradicionales) e incluso más rápidas que las nuevas tecnologías que operan con la línea telefónica tales como ADSL y VDSL. Además, el ancho de banda de las redes Ethernet es *full duplex* en la actualidad, lo que quiere decir que se dispone de todo el ancho de banda desde nuestro ordenador hasta la red y viceversa.

Recientemente, se ha desarrollado un nuevo estándar de conexión que permite alcanzar velocidades de hasta 1.000 Mbps. y que se denomina *Ethernet 1GB*. Si se ha

leído el cuadro anterior titulado “Tipos de cable en una red Ethernet”, uno puede hacerse una idea somera de las características de las diversas clases de redes Ethernet –siendo actualmente *Fast Ethernet* la más usada, aunque en algunas redes locales de alto rendimiento se usa la *Ethernet 1GB*– y del tipo de cableado que se ha de usar con cada clase.

El lector debe observar que no todo tipo de cable sirve para cualquier clase de red, que la topología deberá ser en estrella (la topología en bus ya ha quedado obsoleta), que el elemento central de interconexión será, normalmente, un *switch* (obligado si usamos Ethernet a 1 GB o bien si usamos Fast Ethernet con fibra óptica) y que la distancia máxima a la que se pueden conectar los equipos varía tanto en función de la clase de red como del tipo de cable que empleemos.

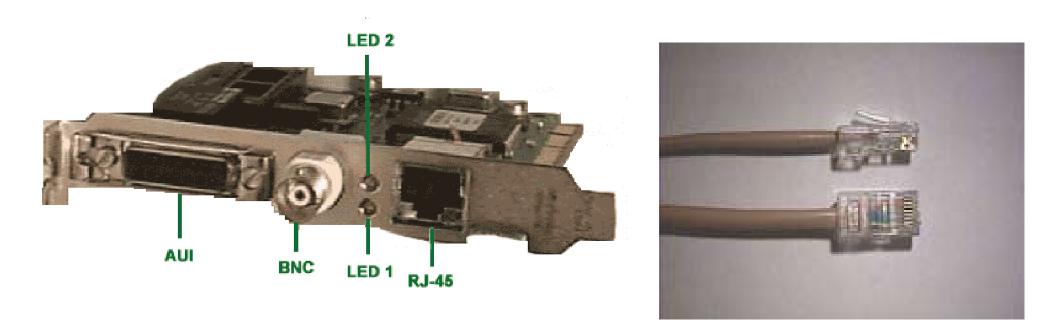


Figura 5.15. A la izquierda se muestran los diferentes conectores que podemos encontrar en una tarjeta de red. A la derecha tenemos una imagen que nos enseña el aspecto que tienen los conectores tipo RJ45

Tal como se ha mencionado anteriormente, las tarjetas de red ofrecen diferentes conectores para su conexión Ethernet en función del soporte físico de la red; fundamentalmente según la topología y el tipo de cable que se emplee. La Figura 5.15 muestra, en su parte izquierda, la imagen de una tarjeta Ethernet con los diversos tipos de conectores destacados en la ilustración. En la actualidad solamente se utiliza el RJ45, ya que habitualmente se usa el estándar *Fast Ethernet*, topología en estrella con *switches* haciendo el papel de elementos centrales y cable de par trenzado. El conector RJ45 permite la conexión a otros dispositivos tales como cable-módems, módems ADSL, *hubs*, *routers*, etc.

No obstante, algunas tarjetas (antiguas) incluyen también el conector denominado BNC, el cual se usaba para la conexión, empleando *cable coaxial*, a una red Ethernet con topología en bus. En este tipo de redes, el cable coaxial constituía una espina dorsal a la que se conectaban todos los equipos. Para poder “engancharse” al cable coaxial (el cable Ethernet) a través de la conexión BNC se empleaban las conocidas “tés”, tal como se ilustra en la Figura 5.16: el cable entraba por un extremo de la “té” y salía por el otro, estableciéndose el contacto con el cable en la parte central de la misma.

El conector AUI (acrónimo del inglés *Attachment Unit Interface*) estaba ya obsoleto a principios de los noventa, por lo que no comentaremos nada más al respecto, ya que se trata de una interfaz de conexión muy antigua.



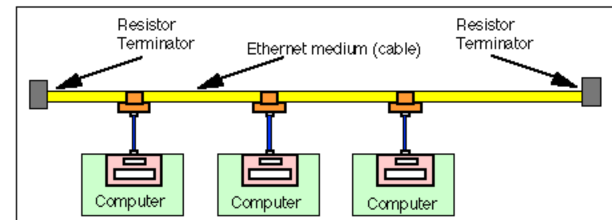


Figura 5.16. A la derecha podemos ver el esquema de conexión a una red Ethernet con topología en bus. A la izquierda tenemos la imagen de una "té", dispositivo empleado para poder establecer la conexión con el cable coaxial sin romper la espina dorsal de la red

Configuración desde Windows y Linux

Ya sabemos que para conectar un ordenador a una red Ethernet se ha de utilizar una *tarjeta de red*. También sabemos que el tipo de conexión varía en función del cable y de la infraestructura (topología) utilizada para "montar" la red.

En la actualidad, podemos encontrarnos con tarjetas que ya vienen integradas en la placa base o bien que se conectan al bus PCI de la misma. Todas las tarjetas soportan el estándar *Fast Ethernet* desde hace ya unos cuantos años y son capaces de funcionar en redes tanto a 10 como a 100 Mbps.

Lo primero que hay que hacer, tanto en Windows como en Linux, para conseguir que funcione la conexión a una red Ethernet es detectar la tarjeta e instalar el programa manejador (*driver*) adecuado en el sistema.

En Windows la detección suele ser inmediata, ya que las tarjetas son *Plug & Play*. En el momento en que el sistema operativo (S.O. en adelante) informa de que ha detectado nuevo *hardware* y que se trata de una tarjeta de red (dando los datos de marca y modelo), pide que se inserte el CD en el que el fabricante proporciona el *driver* para Windows. Normalmente ni siquiera esto será necesario, ya que los *drivers* para los chipsets de las tarjetas más populares ya vienen "de serie" en Windows XP y Vista.

En Linux tampoco suele haber problema. Tanto los kernels de la rama 2.4 como los de la 2.6 incluyen soporte *Plug & Play* para dispositivos PCI y similares. En cualquier caso, se tendrá que cargar el módulo del kernel de Linux adecuado para nuestra tarjeta. Con la mayoría de distribuciones actuales de Linux este proceso será automático, ya que el núcleo de Linux incluye "de serie" los *drivers* de muchas tarjetas de red.

Tras dar soporte al dispositivo en el sistema (sea Windows, sea Linux), habría que configurar, por último, los datos necesarios para que funcione la conexión a la red Ethernet. Estos datos los proporcionará el administrador de la red local (por ejemplo, el Servicio de Informática en la UJI) y, normalmente, son entre otros: dirección IP de nuestro ordenador, nombre de *host*, nombre de *dominio*, direcciones IP para el DNS...

continúa en la página siguiente ...

. . . viene de la página anterior

El propio administrador nos dará instrucciones precisas sobre la configuración tanto en Windows como en Linux, aunque lo más normal es que se emplee el servicio DHCP (más adelante hablaremos de este servicio) para facilitar automáticamente todos estos datos (tanto Windows como Linux ya incorporan programas clientes para DHCP; tan sólo hay que activarlos en la configuración de la red).

Red de cable

A mediados del siglo pasado, surgió la necesidad de llevar señales de televisión y radio, de índole diversa, hasta el domicilio de los abonados, sin necesidad de que éstos tuviesen que disponer de diferentes equipos receptores, reproductores y sobre todo de antenas. Para satisfacer esta necesidad surgieron las primeras *redes de televisión por cable*.

Con el paso del tiempo la tecnología fue evolucionando y estas redes fueron progresivamente aumentando su potencia y alcance. También comenzaron a incluir nuevos servicios con el paso del tiempo, como los de telefonía y transmisión de datos (Internet) aprovechando el ancho de banda sobrante.

Estas redes comenzaron a funcionar en España en los años ochenta. Posteriormente, en los años noventa, fueron desarrollándose: en estos últimos años (finales de los noventa y principios de siglo) se tendieron redes de cable en las ciudades por medio de diversos operadores. Gran parte de estas redes utilizan cable de fibra óptica. En la actualidad, hay varios operadores de cable (siendo Ono el más destacado a nivel nacional) que ofrecen servicios de televisión "a la carta", telefonía e Internet (transmisión de datos y conexión a una red de ordenadores) mediante su propia red de cable. En lo que concierne a las redes de ordenadores, se ofrecen velocidades de transferencia de entre 3 y 50 Mbps. (en descarga) hasta una distancia de 100 km. –luego, en la práctica, la distancia no es un problema.

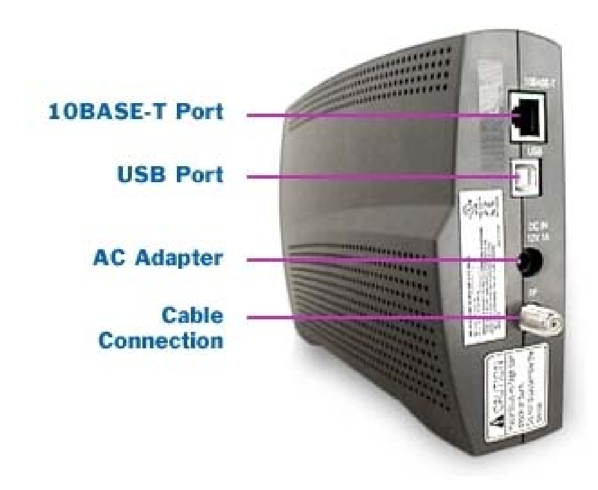


Figura 5.17. Conexiones en la parte trasera de un cable-módem

Para poder conectarse a una red de cable se necesita, como de costumbre, emplear el dispositivo adaptador adecuado. En este caso, se requiere de la utilización de un *cable-módem*. Este dispositivo permite establecer conexiones de *alta velocidad* a Internet a través de una red de televisión por cable. La velocidad de transferencia, tanto en "bajada" (descarga de datos) como en "subida" (envío de datos) estará limitada por el operador que ofrece el servicio[17], aunque en principio los operadores ofrecen conexiones que van desde 2 hasta 50 Mbps. en descarga y desde 128 Kbps. hasta 1 Mbps.

[17] Esto suele realizarse mediante un fichero de configuración que el operador escribe en la "memoria"

de "subida". Al usar cableados distintos para voz y datos, son compatibles con la línea telefónica, por lo que es posible enviar y recibir llamadas mientras estamos conectados a la red. Cabe resaltar que la red de cable no tiene por qué ser asimétrica en principio (al contrario que ADSL): la asimetría es más bien el resultado de la forma que tiene el proveedor de repartir el ancho de banda que proporciona su red.

Podemos encontrar cable-módems internos (son tarjetas que se conectan a una ranura libre del bus PCI de la placa base de nuestro PC, aunque hoy en día son raras de encontrar y la mayoría están descatalogadas) o externos, que es lo más habitual. La Figura 5.17 muestra una ilustración de las conexiones que podemos encontrar, normalmente, en la parte trasera de un cable-módem externo. Sea interno o externo, el cable-módem se ha de conectar a la conexión que el operador de cable ofrece en nuestro domicilio para su red ("Cable Connection" en la figura). De hecho, la única restricción es que el cable-módem se ha de conectar, en último término, a un sistema de terminación especial que deberá proporcionar el proveedor del cable en su central de conexión.

Por otro lado, si el cable-módem es interno tan sólo tenemos que instalar el *driver* correspondiente para nuestro sistema operativo y configurarlo según indique el proveedor. Por tanto, para poder operar con el cable-módem es necesario que esté soportado por nuestro sistema. En el caso de tener un cable-módem externo su conexión con el PC puede realizarse de dos maneras distintas:

- Directamente, a través de un puerto USB (la mayoría de estos aparatos admite conexión directa a USB): véase la conexión "USB Port" en la Figura 5.17. En este caso, debemos instalar el *driver* correspondiente a nuestro sistema operativo para que podamos operar con él. Lo normal es que, debido a las características de "conexión en caliente" (*hotplug*) del bus USB, el cable-módem sea detectado al enchufarlo y entonces se cargue el *driver* (la primera vez, en el caso de Windows, puede ser que el sistema solicite la introducción del CD del fabricante con el *driver* para copiarlo al disco duro). Nuevamente, para que funcione es necesario que esté soportado en nuestro sistema. En el caso de Linux, debemos asegurarnos de que el *driver* ya está incluido en el kernel de nuestra distribución, o bien instalarlo previamente (suele haber instrucciones bastante precisas al respecto).
- Indirectamente, mediante una conexión Ethernet a una tarjeta de red que deberíamos tener instalada en nuestro PC (empleando cable de par trenzado): véase la conexión "10BASE-T Port" en la Figura 5.17. Por tanto, tan sólo hace falta que la tarjeta de red esté soportada en nuestro sistema y no el cable-módem. Esta opción es la que menos problemas nos dará en Linux, ya que existe un mayor soporte para una gran variedad de tarjetas de red (y puede que no sea el caso de que el cable-módem esté soportado). En este caso, realmente se crea una mini-red local (LAN) entre nuestro PC y el cable-módem que hace el papel de *enrutador* (*router*) hacia la red de cable del proveedor.

---

del cable-módem en el momento de establecer la conexión. El usuario puede borrar ésta y establecer su propia configuración, saltándose las limitaciones del operador y accediendo a un mayor ancho de banda de la red, pero, según la legislación vigente en nuestro país, debes saber que esta acción constituye un delito.

A la tecnología y al protocolo de comunicaciones que se utiliza en una red de cable (y que debe soportar el cable-módem) se les conoce con el nombre de DOCSIS (acrónimo del inglés *Data Over Cable Service Interface Specification*) y tiene varias versiones. La más antigua (y ya obsoleta) fue la 1.0. En la actualidad se emplea la 1.1 y la 2.0. El operador de cable Ono ha estado realizando pruebas recientemente con la versión 3.0, la cual permite alcanzar velocidades superiores a los 100 Mbps. El lector debe tener presente que el cable-módem que use debe ser compatible con la versión de DOCSIS que esté utilizando el operador de cable con el que desea contratar el servicio[18].

Wireless

Las redes inalámbricas, conocidas popularmente por su nombre inglés *Wireless* (literalmente "sin cable") han experimentado un auge considerable en los últimos años y, en la actualidad están presentes en los hogares, en la universidad, en bibliotecas, en empresas, etc. En realidad, el término *Wireless* es un diminutivo de la expresión más general que identifica este tipo de redes: WLAN, acrónimo de *Wireless Local Area Network*.

La idea básica consiste en transmitir datos a través del aire, mediante ondas hertzianas, es decir ondas de radio y microondas (radiofrecuencia). La ventaja es que no se requiere de ningún tipo de cableado para montar la red. El lector perspicaz se habrá percatado de que las desventajas tienen que ver fundamentalmente con los problemas planteados por la *distancia* de emisión (atenuación de las ondas), *interferencias* (producidas por otras ondas en frecuencias similares o incluso por fenómenos atmosféricos, como la lluvia) y la *seguridad en las comunicaciones* (para acceder a los datos en una red cableada hay que "pinchar" el cable o bien acceder a un equipo conectado a la misma, mientras que para acceder a una red inalámbrica basta, en principio, con poner una antena y "sintonizar" la frecuencia correspondiente).

El protocolo estándar de comunicaciones que define el funcionamiento de este tipo de redes es el *IEEE 802.11* y entre los dispositivos (*hardware*) que se requieren para su correcto funcionamiento encontramos tarjetas de red inalámbricas y *hubs*, *switches* y *routers* inalámbricos.

Las redes inalámbricas fueron inicialmente ideadas para establecer pequeñas redes, como por ejemplo la que puede dar cobertura a una planta de oficinas en un edificio. Sin embargo, debido a su alta popularidad y comodidad de uso –desde el punto de vista tanto del usuario como de la infraestructura que hay que mantener– se están extendiendo cada vez con mayor rapidez en el mundo empresarial volviéndose especialmente útiles en pequeñas y medianas empresas, ya que así se ven libres de cableados e instalaciones particulares.

Para su utilización práctica en extensiones mayores de terreno, como la que presenta un campus universitario, por ejemplo, se emplean antenas con un alcance entre 2 y 8 kilómetros. Éstas proporcionan el soporte necesario para efectuar las comunicaciones en el área establecida. Hoy en día las redes inalámbricas pueden abarcar no sólo todo

[18]En el caso de que lo desee adquirir por su cuenta, lo cual casi siempre resulta más barato que quedarse con el modelo ofrecido por el proveedor.

un campus universitario (como el de la UJI) sino también ciudades enteras[19], como fue el caso de Zamora en el año 2002.

En su momento, las redes inalámbricas plantearon serios problemas legales. Puesto que estas redes emiten en bandas públicas de frecuencia, no se requieren licencias para trabajar en ellas. Por ello, algunas comunidades de usuarios vieron la oportunidad para dar acceso gratuito a Internet o bien compartir conexiones de banda ancha entre varios usuarios. La Comisión del Mercado de las Telecomunicaciones (CMT), organismo encargado de velar por la libre competencia en el mercado de las telecomunicaciones y por el interés general[20], ha ido obligando a eliminar el uso de redes públicas para el acceso directo a Internet.

Para poder ofrecer un servicio de comunicaciones (como por ejemplo el acceso a Internet) la comunidad o grupo de usuarios (o el ayuntamiento correspondiente si fuera el caso) debe constituirse en operador y eliminar toda financiación pública (se supone que para evitar la competencia desleal con los operadores privados que ofrecen servicios de pago para la conexión a Internet), es decir, debe convertirse en otro operador privado. Ello supone, en la práctica, la prohibición de estas redes. La única excepción a esta norma es que la CMT considere la red establecida como de *interés público*. De momento, los autores de estos apuntes no conocemos de ningún caso donde se haya considerado así una red pública con acceso sin restricciones a Internet.

Por otro lado, en estos casos se debe tener en cuenta que resulta de aplicación la Ley General de Telecomunicaciones (LGT) y, de hecho, los requisitos mencionados en el párrafo anterior están establecidos en dicha ley. En lo que se refiere a la compartición de la conexión a Internet (con el vecino, por ejemplo), se ha de tener en cuenta la reciente reforma del código penal en España (año 2004), en la que se indica que es delito compartir la conexión a Internet con fines comerciales sin contar con la autorización del proveedor de la conexión. Aun suponiendo que no se realizase con fines comerciales (o que esto no pudiese ser probado) quien realice este tipo de actividades sin permiso del proveedor se expone a una fuerte multa. Se recomienda, en cualquier caso, leer detenidamente el contrato con el proveedor, aunque la mayoría de proveedores desautorizan estas actividades ya en el mismo.

En lo que concierne a los problemas de seguridad, quisiéramos destacar que en las redes inalámbricas existen graves lagunas que son fácilmente explotables por usuarios con conocimientos de redes. Como ya hemos dicho anteriormente, la difusión de los datos en bandas públicas de radiofrecuencia hace que éstos se puedan interceptar fácilmente (basta poner la antena –ergo una tarjeta inalámbrica de red– a "escuchar" en la frecuencia adecuada). En particular, si los puntos de acceso a la red inalámbrica no están "protegidos" (o no lo están suficientemente) resulta sencillo conectarse a ellos y acceder a la red correspondiente sin autorización. Valga como ejemplo de estas prácticas el denominado *wardriving*: consiste en ir circulando con un coche por la calle llevando un portátil equipado con un dispositivo (tarjeta de red) inalámbrico y buscando conexiones "abiertas" o poco protegidas.

Obviamente, el consorcio que desarrolló el estándar IEEE 802.11 conocía los problemas de seguridad que podrían darse, por lo que acompañó la definición del estándar

[19] Estamos hablando, no obstante, de pequeñas poblaciones.

[20] Se supone.

de la especificación de un sistema de cifrado conocido como WEP (acrónimo del inglés *Wired Equivalent Privacy*). El problema con WEP es que, sobre el año 2001, reputados analistas criptográficos identificaron varias debilidades en dicho protocolo de seguridad. En la práctica, ello hace que hoy en día una protección WEP pueda ser violada con programas fácilmente accesibles –en Linux, por ejemplo, tenemos el programa `airsnort` disponible como paquete en muchas distribuciones, véase `http://airsnort.shmoo.com/` para más detalles– en pocos minutos (si se trata de una clave corta) o unas pocas horas (si se trata de una clave más larga).

Poco tiempo después de descubrirse estas vulnerabilidades, el comité correspondiente del IEEE creó una nueva corrección de seguridad, la denominada 802.11i, para neutralizar los problemas del WEP. En el año 2003, el comité Wi-fi anunció que WEP había sido reemplazado por el protocolo denominado *Wi-fi Protected Access*, popularmente conocido por sus siglas WPA. A partir de ese momento, se comercializaron nuevos *routers* y *switches* que lo soportaban, aunque tardaron un tiempo. A pesar de que aún no se había concluido con la especificación del estándar 802.11i, resultaba urgente poner en marcha nuevas extensiones específicas de seguridad debido a la fragilidad del WEP.

Finalmente, en 2004, con la ratificación del estándar definitivo 802.11i, en adelante conocido como WPA2 para diferenciarlo de su primera versión (WPA), el comité Wi-fi declaró que WEP había sido descartado por su fracaso en alcanzar los propósitos de seguridad perseguidos. La mayoría de *routers*, *switches* y dispositivos inalámbricos en general comercializados a partir de entonces soportan la especificación completa del estándar 802.11i, es decir, que soportan WPA2.

El lector debe tener en cuenta que, a pesar de sus debilidades, WEP sigue siendo bastante utilizado, ya que es a menudo la primera opción de seguridad que las herramientas de configuración de los *routers* inalámbricos "caseros" presentan a los usuarios, la mayoría legos en asuntos informáticos, por lo que desconocen que WEP tan sólo proporciona un nivel de seguridad que puede disuadir del uso sin autorización de una red privada, pero sin proporcionar una verdadera protección frente a usos no autorizados. A pesar de todo, WEP todavía está documentado en el estándar actual. Se recomienda encarecidamente al lector que, si dispone o administra una red inalámbrica y desea protegerla realmente, active las opciones de seguridad WPA o, mejor aún, WPA2.

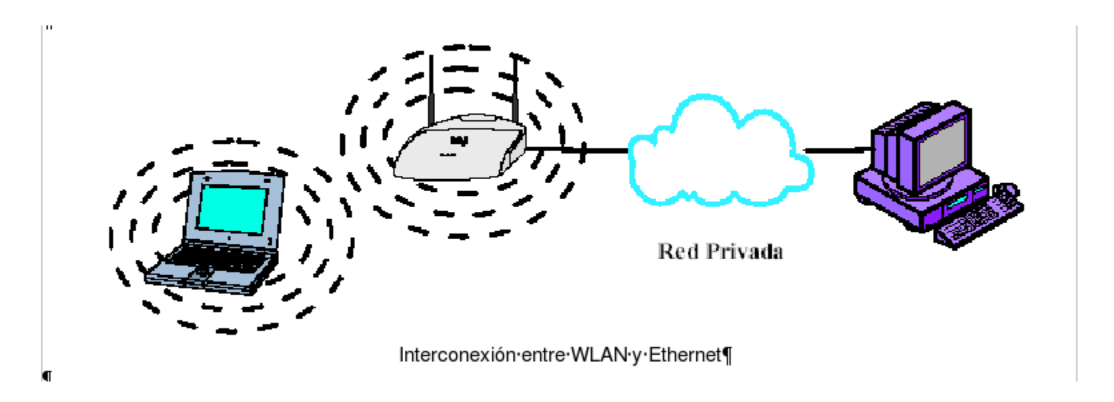


Figura 5.18. Conexión entre una red inalámbrica y una Ethernet tradicional

Desde el punto de vista de las comunicaciones propiamente dichas (transferencia de datos), normalmente las redes inalámbricas locales (WLAN, *Wireless Local Area Network*, en adelante) se usan como una extensión de las redes Ethernet tradicionales, aunque la comunicación sobre ellas puede ser realizada en modo "punto-a-punto" –es decir, directamente de ordenador a ordenador.

Debe tenerse en cuenta que, para que una WLAN pueda funcionar en consonancia con una red Ethernet es necesario instalar y configurar adecuadamente un *punto de acceso* o *bridge* (puente, traducido del inglés) que haga de nexo y traductor entre ambas redes. La Figura 5.18 muestra gráficamente este esquema de conexión, donde el punto de acceso será habitualmente un enrutador (*router*) inalámbrico. Este esquema es el que se emplea en la red inalámbrica de la UJI, donde para acceder a la red propia se ha de estar situado cerca de uno de los puntos de acceso establecidos (hay varios estratégicamente repartidos por todo el campus, dando cobertura a gran parte de la superficie universitaria).

Hay que tener en cuenta que, debido a que se utilizan ondas de radio como medio de comunicación, la potencia de la señal, que afecta a la velocidad de transmisión, variará en función de la distancia a la que nos encontremos del punto de acceso, las interferencias que haya en el ambiente, etc.

Para que un PC equipado con una tarjeta inalámbrica pueda acceder a una red determinada, es necesario que ésta pueda identificarse de algún modo. Para ello, los puntos de acceso difunden un mensaje en el que anuncian (entre otras cosas, como, por ejemplo, el mecanismo de seguridad empleado) el identificador (SSID) de la red sobre la que operan. El SSID (acrónimo del inglés *Service Set IDentifier*) es un código incluido en todos los paquetes de una red inalámbrica para identificarlos como parte de esa red. Todos los dispositivos inalámbricos que intentan comunicarse entre sí deben compartir el mismo SSID. Este código permite a los dispositivos inalámbricos conectarse al punto de acceso que lo ha difundido, pudiendo acceder del mismo modo a los servicios Ethernet a que se encuentra conectado.

Es importante resaltar que los puntos de acceso pueden operar en diferentes *canales* dentro de una misma banda de frecuencias. Esto puede contribuir a solucionar las interferencias entre dos puntos de acceso que cubren una misma zona (normalmente, casi siempre existen solapamientos, pero sólo en algunas ocasiones se producen interferencias). Si existe solapamiento y hay interferencias, el problema puede solucionarse simplemente cambiando de canal. También podemos emplear filtros específicos para solucionar este tipo de interferencias, en el caso de que no queramos cambiar el canal, pero hay que advertir de que son caros.

En lo que se refiere al uso de protocolos estándar para efectuar las comunicaciones existen diversas variaciones dentro de lo que podríamos denominar la "familia" 802.11. De hecho, esta familia actualmente incluye seis técnicas de transmisión por modulación que utilizan todas los mismos protocolos. Destacaremos, entre ellos, cuatro estándares de transmisión, siendo actualmente los dos últimos los más utilizados:

(a) *802.11 legacy* (año 1997). Se trata de la versión original de este estándar IEEE. Permitía operar con un ancho de banda máximo de 2 Mbps., trabajando en la banda de frecuencias de 2,4 GHz. La aparición de dificultades en la comunicación entre dispositivos de diferentes marcas y otros problemas técnicos contribuyeron

a su obsolescencia. Estos problemas fueron corregidos en la versión *802.11b* del protocolo. Actualmente ya no hay implementaciones disponibles del estándar original.

(b) *802.11a* (año 1999). Ésta es una versión de alta velocidad del estándar, ya que permite un ancho de banda máximo de 54 Mbps., aunque en promedio lo habitual sea obtener entre 25 y 40 Mbps. en función de la distancia al punto de acceso, de las interferencias, etc. Esta versión del protocolo opera en la banda de frecuencias de 5 GHz. Tiene un alcance interior de unos 30 metros para su funcionamiento a plena velocidad. Al operar en la banda de 5 GHz. presenta menos interferencias que el original, pero, en contrapartida, necesita más puntos de acceso y sufre mayores problemas de atenuación de la señal. Por motivos técnicos casi no se desarrollaron productos, por lo que su uso no se extendió en demasía, siendo, además, incompatible con los productos desarrollados bajo la especificación *802.11b*.

(c) *802.11b* (año 1999). Este estándar sí que fue muy exitoso y, de hecho, es el que da su nombre popular al resto de la "familia", al ser conocido como *Wi-Fi* (por el inglés *Wireless Fidelity*). Actualmente es el más extendido, en pugna con el cada vez más usado *802.11g*. En realidad fue una revisión menos drástica que la anterior, ya que también funciona en la banda de frecuencias de 2,4 GHz., tal como el original. Sin embargo, permite una velocidad máxima de transferencia de 11 Mbps. siempre que estemos a una distancia del punto de acceso no superior a los 50 metros.

(d) *802.11g* (año 2003). La especificación *802.11g* permite obtener altas velocidades de transferencia operando en la banda de frecuencias de 2,4 GHz, por lo que podría ser compatible con *802.11b*. Y, de hecho, lo es. En principio, se puede obtener un ancho de banda máximo de 54 Mbps., a una distancia de un máximo de 30 metros del punto de acceso. En promedio, se pueden obtener velocidades entre 25 y 40 Mbps. en función de la intensidad de la señal, interferencias, etc. También se le denomina popularmente *Wi-Fi*. Aunque es compatible con los dispositivos *802.11b*, la velocidad se reduce hasta los 11 Mbps. máximo cuando se comunica con ellos (para poder ser compatible).

En cuanto a la seguridad, ya hemos hablado de ella anteriormente. Tan sólo queremos resaltar que ha formado parte de la especificación del protocolo desde el principio (WEP), y que fue mejorada en la revisión *802.11i*. Desde el año 2005, la mayoría de los productos que se comercializan siguen el estándar *802.11g* con compatibilidad hacia el *802.11b*.

Estaba previsto, para finales de 2006 o principios de 2007, la aparición de una nueva especificación mucho más potente y funcional: la denominada *802.11n*. Sin embargo, debido a problemas de diversa índole todavía no se ha alcanzado la redacción definitiva del documento que describe las características del estándar. Se espera que el borrador definitivo aparezca a finales de 2009. Esta especificación permitirá operar en las bandas de frecuencias tanto de 2,4 como de 5 GHz y alcanzar una velocidad máxima de 600 Mbps. a una distancia máxima de 70 metros del punto de acceso. Actualmente ya existen varios productos en el mercado que cumplen un primer borrador

de este estándar y que ofrecen un máximo de 300 Mbps. (en la práctica entre 80 y 100 estables), con promesa de actualizaciones del *firmware* para cuando salga la versión definitiva del protocolo.

Puesto que los estándares *802.11b* y *802.11g* utilizan las bandas de frecuencia de 2,4 GHz., que no necesitan de permisos para su uso, es posible que se padezcan interferencias por parte de hornos microondas, teléfonos inalámbricos, dispositivos equipados con *bluetooth* y otros equipos que utilicen la misma banda de 2,4 GHz. Ésta es una de las mayores fuentes de problemas, junto con los problemas de seguridad ya mencionados y los problemas de atenuación de la señal por la distancia o bien por condiciones atmosféricas adversas, como por ejemplo la lluvia abundante.

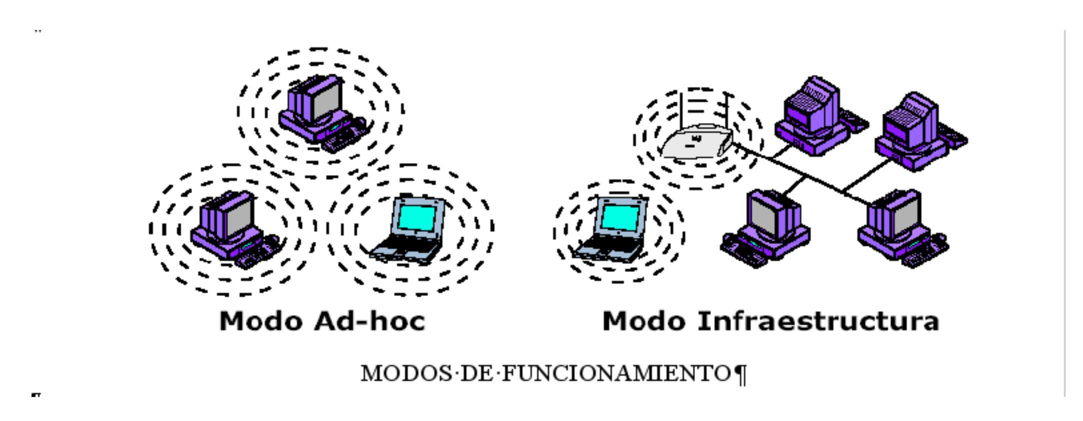


Figura 5.19. Topologías de funcionamiento de las redes *Wireless*

En cuanto a la topología o forma de conexión, existen dos maneras de funcionar, ilustradas en la Figura 5.19: la topología *ad-hoc*, donde cada dispositivo actúa de forma *autónoma* mediante comunicación directa o "punto-a-punto"; y el modo (topología) *infraestructura*, mucho más frecuente, y que sigue el modelo cliente-servidor, siendo los puntos de acceso los nodos principales que difunden la información a todos los terminales conectados.

Otros

Aunque los soportes de red (o medios físicos, como prefiera el lector) que hemos tratado en las secciones anteriores son los que se han usado, se usan y se usarán –en los próximos años– con mayor frecuencia para la conexión de redes (y, por tanto, el acceso a Internet), vale la pena comentar algunas cuestiones al respecto de otras posibilidades existentes.

En particular, queremos mencionar que también cabe la posibilidad de conectar equipos de forma remota empleando otro tipo de cableado ubicuo en nuestra vida cotidiana: *el cable de la luz*. En las ciudades y pueblos (grandes y pequeños) existe el tendido eléctrico, el cableado que permite que la electricidad llegue hasta nuestros hogares. Pues bien, la tecnología PLC (acrónimo del inglés *Power Line Communications*) permite ofrecer conexión a Internet a través del cable de la luz que llega hasta nuestra casa. Veamos sus características más destacadas:

- Puesto que la infraestructura (el tendido eléctrico) ya existe, no es necesario tender cable para montar la red. La compañía eléctrica propietaria del tendido

eléctrico tiene ya montada buena parte de la infraestructura necesaria. Sin embargo, la compañía debe instalar aparatos receptores especiales en las centralitas eléctricas, y estos aparatos son bastante caros (por lo que es necesaria una cierta inversión). Desde el punto de vista del usuario, basta con instalar un *filtro específico* en el enchufe donde se vaya a realizar la conexión del computador y usar un *módem específico* (PLC).

- El ancho de banda máximo que puede ofrecerse es bastante competitivo con el ADSL actual: hasta 20 Mbps. Pero no puede decirse lo mismo si comparamos con ADSL2 y las tecnologías VDSL y DOCSIS 3.0 (red de cable).

- No obstante, la tecnología PLC presenta una ventaja notoria, y es que no es una tecnología meramente urbana, ya que el tipo de enchufe empleado es el convencional, es decir, que donde llegue la electricidad llega Internet, y de alta velocidad. Esto no lo consigue el ADSL actual ni las redes de cable, que no suelen ofrecerse en los pueblos (o si se ofrecen no resultan muy atractivas debido a las distancias hasta la centralita) ya que a las "telecos" no les compensa la inversión.

- Sin embargo, parece que los inconvenientes superan a las ventajas, ya que si al hecho de que los equipos receptores especiales son muy caros le sumamos la ya amplia implantación del cable y el ADSL, nos encontramos con que las compañías eléctricas sólo ofrecen servicio en una zona si piensan que les puede resultar rentable. Por ejemplo, la compañía eléctrica Iberdrola –concesionaria de la tecnología en nuestro país– nunca terminó de apostar fuerte por el PLC. En su momento llegó a ofrecerlo en algunas zonas urbanas (Zaragoza, por ejemplo), pero nunca le dieron demasiada publicidad y hasta parecían tenerlo "escondido" en su página web. Actualmente, Iberdrola ya ha abandonado esta tecnología[21].

También vale la pena mencionar que existe una solución para el acceso "ocasional" a Internet, normalmente a través de un portátil, y desde cualquier punto: *las tarjetas de telefonía 3G*. Estas tarjetas operan en realidad de forma similar a las de los teléfonos móviles: de hecho, pueden considerarse como teléfonos móviles. La tarjeta se conecta al portátil mediante un puerto PCMCIA (específico de los portátiles) y, al ponerse en funcionamiento, se conecta a la red UMTS/3G del operador correspondiente, proporcionando en teoría hasta 2 Mbps. (se supone que son simétricos).

Para su correcto funcionamiento es necesario que en el lugar donde nos encontremos haya cobertura para la red UMTS/3G del operador al que hemos contratado la tarjeta. Cuanto mayor sea la cobertura menos "cortes" experimentaremos y la velocidad de transferencia sostenida será mayor. Muchas tarjetas funcionan de modo que si no hay cobertura entonces pasan al modo GPRS para el que (casi) siempre existe cobertura, aunque en este caso el ancho de banda será mucho menor.

También se puede usar otro adaptador para la conexión a una red de telefonía UMTS/3G: un *módem 3G USB*. Este dispositivo se conecta al puerto USB del or-

[21] Las "malas lenguas" dicen que PLC estaba condenado al fracaso en nuestro país, ya que Iberdrola es (o lo era en su momento) accionista en varias "telecos" que ofrecen ADSL.

denador (portátil) y, mediante las opciones de configuración que indica el operador[22], se conecta a su red 3G. El ancho de banda máximo que se puede obtener, en teoría, con este tipo de conexión varía en función del operador y la cobertura de su red 3G; aunque puede estar en torno a los 6 ó 7 Mbps. (simétricos). Según planes y ofertas de los operadores, lo normal es tener velocidades entre 1 y 4 Mbps.

Casi todas las compañías de telefonía móvil que operan en nuestro país (que sepamos, al menos están Orange, Movistar y Vodafone) ofrecen este servicio. El lector interesado debería consultar las ofertas disponibles, aunque cabe destacar que en la mayoría de ellas se paga por volumen de datos transmitidos.

5.4.2. Concentradores

Un concentrador, popularmente conocido por su denominación en inglés (*hub*), es un dispositivo básico para la comunicación en las redes de ordenadores que emplean la denominada *topología en estrella* (la más frecuente), y que permite conectar entre sí varios ordenadores. En este sentido, realiza un papel similar al que jugaba el cable coaxial empleado en las (antiguas) redes con topología en bus (constituye igualmente la "espina dorsal" en las comunicaciones). De hecho, el *hub* actúa de modo que todos los cables están conectados entre sí. En la parte izquierda de la Figura 5.20 podemos ver el aspecto que tiene un *hub*.

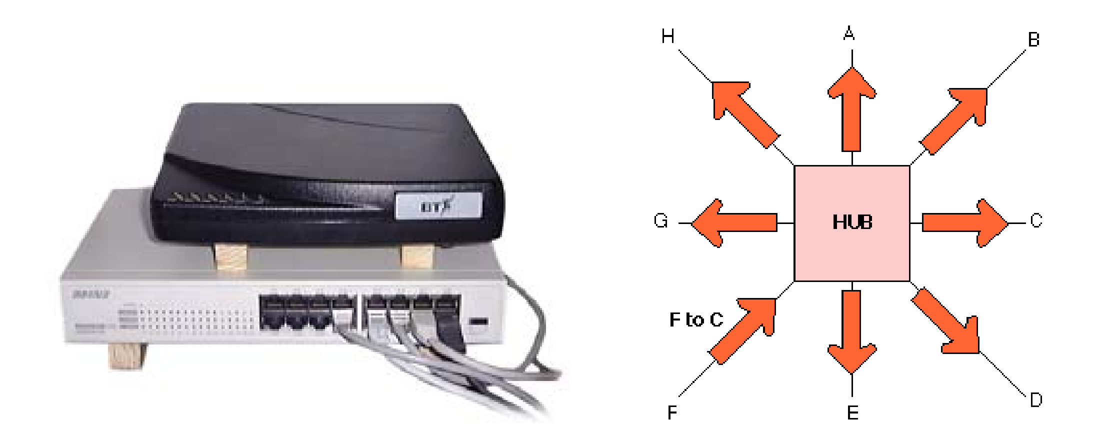


Figura 5.20. A la izquierda tenemos la imagen de un *hub*. A la derecha observamos una ilustración que indica cómo se desarrollan las comunicaciones entre los equipos conectados al mismo

Cuando un ordenador se conecta al *hub*, éste se encarga de establecer y posibilitar su comunicación con otros ordenadores de la red. Esto lo consigue retransmitiendo los datos hacia todos los ordenadores conectados (lo que se envía a la red es leído por todos), tal como se indica en la parte derecha de la Figura 5.20.

Los *hubs* permiten la posibilidad de interconectar diversas redes entre sí, ya que ellos mismos se pueden interconectar unos con otros, y por ende las redes que cada uno enlaza, para ir formando redes cada vez más amplias, tal como indica la Figura 5.21.

[22]Normalmente, el operador facilita un programa que ya se encarga de configurar automáticamente el dispositivo en el momento de conectarlo al puerto USB. Desafortunadamente, ese hipotético programa suele estar disponible sólo para sistemas operativos de la "familia" Windows.

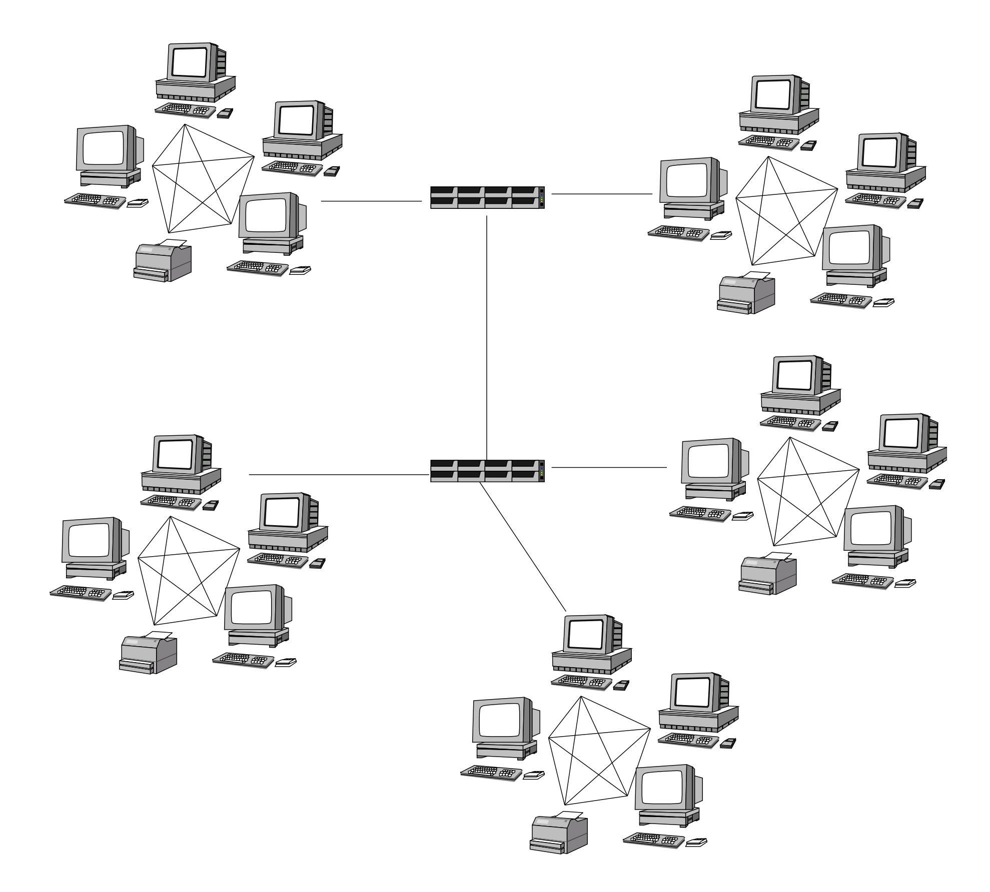


Figura 5.21. La interconexión de dos *hubs* permite, igualmente, la interconexión de las redes que éstos enlazan

En la actualidad los *hubs* tienen un papel menos protagonista, ya que han sido desplazados con frecuencia de su rol de elemento central para la conexión de ordenadores en redes con topología en estrella por los *switches* (conmutadores). Los *switches* son más "inteligentes" (en el sentido de que se les puede "programar" para realizar tareas mucho más sofisticadas, como por ejemplo, filtrar el tráfico de la red mediante el análisis de las cabeceras de los paquetes) y permiten que se mantengan varias conexiones simultáneamente.

Hasta hace unos años la forma más sencilla de interconectar dos o más ordenadores domésticos era empleando un *hub* al que se conectaban las tarjetas de red de cada ordenador. Ello requería "tirar" cable por la casa, pero menos que usando una topología en bus (y, además, era mucho más funcional y fácil de configurar). Con el tiempo, esta posibilidad de uso del *hub* se ha visto desplazada por la tecnología *wireless*, que constituye actualmente la mejor alternativa para montar una red casera y compartir el acceso a Internet: no se requieren cables y basta con adquirir un *router* o un *switch* inalámbrico y configurarlo adecuadamente.

5.4.3. Equipos de ampliación e interconexión de redes

Para lograr un funcionamiento adecuado de las redes de ordenadores, deben utilizarse dispositivos "complementarios" que contribuyen a solucionar los problemas que se plantean en las comunicaciones de los equipos conectados a medida que aumenta el

tráfico y el tamaño de la red y la distancia de conexión.

En esta sección, queremos comentar brevemente las características de los dispositivos más conocidos e importantes con el fin de proporcionar al lector una idea somera de su utilidad y capacidades: repetidores, puentes (*bridges*), conmutadores (*switches*), encaminadores (*routers*) y pasarelas (*gateways*).

Repetidores

Los repetidores son dispositivos sencillos que, básicamente, amplifican la señal que llega por el cable para que así pueda alcanzar mayores distancias. De paso, incluyen filtros que permiten eliminar o, al menos, paliar en parte algunos de los problemas más comunes en las transmisiones, tales como los ocasionados por la distorsión de la señal y la presencia de ruido (debido a interferencias, etc.), y que se ejemplifican gráficamente en la parte superior de la Figura 5.22.

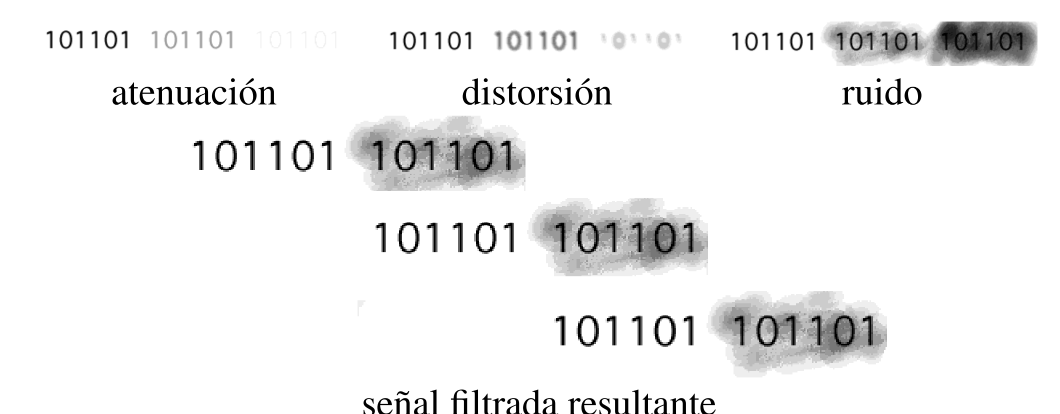


Figura 5.22. Arriba tenemos ejemplos gráficos de lo que ocurre con los bits en las transmisiones de datos: la señal se atenúa (los bits se "difuminan" hasta desaparecer), se distorsiona (hasta que no se sabe muy bien qué valor tienen los bits) y se "ensucia" por el ruido. Abajo, la señal saliente amplificada y filtrada (queda algo de ruido, pero no lo suficiente para perturbar la señal)

Los repetidores suelen constar de una única entrada y de una única salida, a través de las cuales circula la señal que se pretende filtrar, amplificar y, en cierto modo, reconstruir, tal como se ilustra en la parte inferior de la Figura 5.22. Aunque también son de utilidad actualmente, en su momento era bastante frecuente encontrarlos en redes con topologías en bus, con el propósito de amplificar la señal que circulaba por el cable coaxial y mejorar el rendimiento de la red al producirse menos retransmisiones de paquetes. La Figura 5.23 muestra un ejemplo de este uso.

Puentes

Los puentes son dispositivos de interconexión de redes de ordenadores que operan a nivel de enlace de datos y que se utilizan, fundamentalmente, para interconectar dos segmentos de red distintos haciendo el pasaje de datos de una red hacia otra y que actúen como si fuesen una única red. Por tanto, permiten agrupar dos redes para formar otra de mayor extensión. La Figura 5.24 muestra cómo funciona conceptualmente un puente.

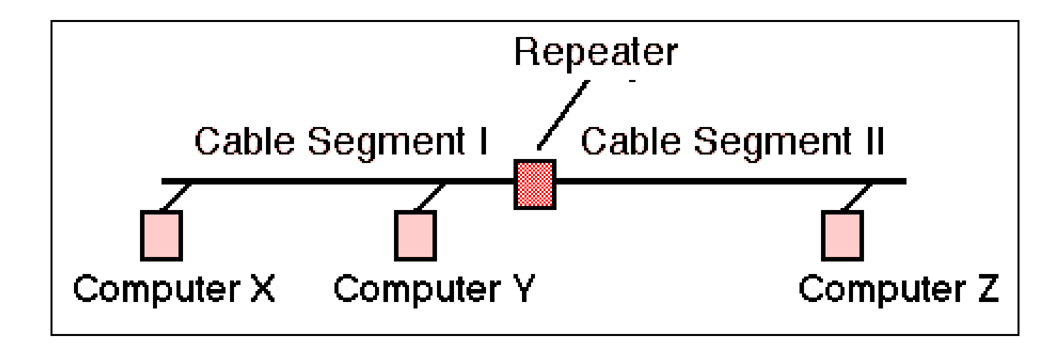


Figura 5.23. Un repetidor amplifica la señal que circula por el cable coaxial que constituye la "espina dorsal" de una red con topología en bus

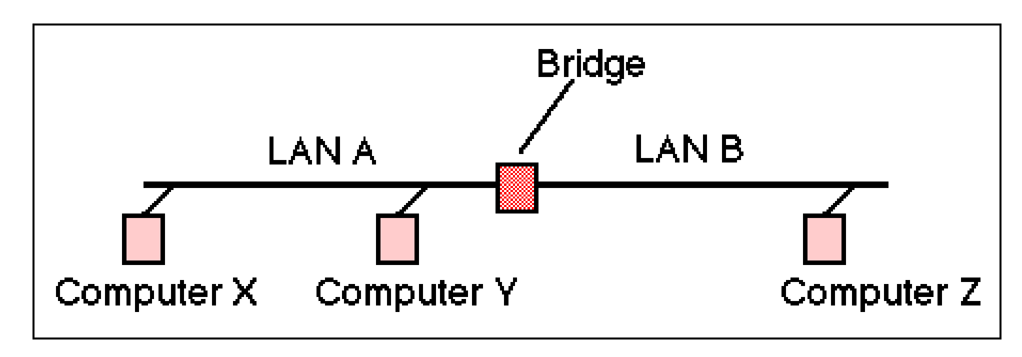


Figura 5.24. Un puente permite unir, en una única red, dos segmentos distintos

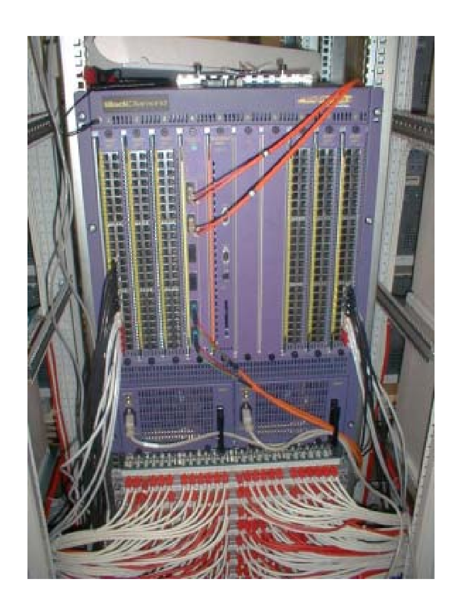
Curiosamente, el otro uso que se les ha dado a estos dispositivos consiste en dividir (y conectar) una red ya existente en dos segmentos de red más pequeños, con el fin de aumentar el rendimiento global al disminuir el número de colisiones en la misma y mejorar el tráfico. Esto es así debido a que cuando existe un gran número de ordenadores conectados a una misma red entonces aumenta el número de colisiones. El puente permite separar el tráfico interno de cada segmento de red y evita muchas de ellas.

La principal diferencia entre un puente y un *hub* es que el segundo pasa cualquier paquete de datos con cualquier destino a todos los otros nodos conectados, mientras que el primero sólo pasa los paquetes pertenecientes a cada segmento. Como hemos dicho, esta característica mejora el rendimiento de las redes al disminuir el tráfico inútil. Cuando se desea interconectar más de dos redes se utilizan los *switches* o conmutadores.

Conmutadores

Un conmutador, popularmente conocido por su denominación inglesa *switch*, es un dispositivo similar en su utilidad a un *hub*, en el sentido en que sirve para conmutar líneas y repartir el tráfico entre todos los equipos que se le conectan. Igualmente, un *switch* se utiliza como dispositivo central (eje de las comunicaciones) en las redes con topología en estrella. En la parte izquierda de la Figura 5.25 podemos apreciar el aspecto que tiene un *switch*.

A la hora de establecer la comunicación entre dos ordenadores diferentes, el *switch*,



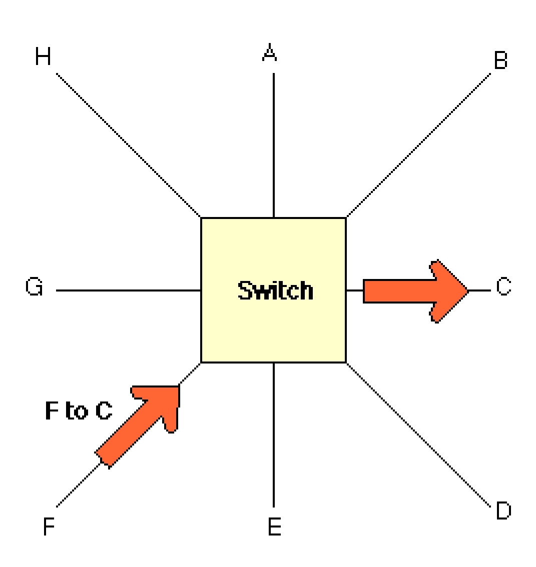


Figura 5.25. A la izquierda podemos ver un *switch* con todas las conexiones que llegan a él. A la derecha observamos una ilustración que indica cómo se desarrollan las comunicaciones entre los equipos conectados al mismo

a diferencia del *hub*, que retransmite los datos hacia todos los ordenadores conectados –y, por tanto, lo que se envía a la red es leído por todos–, sólo transmite los datos entre los ordenadores implicados, tal como se indica en la parte derecha de la Figura 5.25. Esto significa que los *switches* son capaces de establecer conexiones "dedicadas" entre dos nodos de la red. Esta característica permite que los *switches* reduzcan la carga en la red en comparación con la generada por los *hubs*.

Como ya comentamos en la sección 5.4.2 al hablar de los *hubs*, los conmutadores les han postergado en buena medida, ocupando su lugar central en las topologías de conexión en estrella. Aparte de la posibilidad de establecer conexiones "dedicadas" y de gestionar varias conexiones de manera simultánea, esto se ha producido por el hecho de que los *switches* añaden una cierta "inteligencia" en sus funciones que no poseen los *hubs*. Por ejemplo, los *switches* son programables, permiten especificar filtros para los diferentes puertos TCP (ya veremos qué es esto más adelante), posibilitan la definición de reglas para el filtrado de ciertos tipos de paquetes de datos (datagramas) que circulan por la red, etc. En la actualidad, también existen *switches* inalámbricos para su uso en redes *wireless*.

Encaminadores

Un encaminador, enrutador o, sencillamente, *router* (tomado tal cual del inglés) es un dispositivo que se emplea, fundamentalmente, para realizar interconexiones de redes. Desde el punto de vista de la gestión del tráfico de la red, se podría comparar con un *hub* o un *switch*, aunque en principio los *routers* suelen ser equipos más sofisticados.

Básicamente, los *routers* son los "mensajeros" de Internet. Ellos se encargan de dirigir el tráfico de datagramas por las rutas correctas hasta llegar a su destino. Conocen las posibles rutas que pueden existir para llevar un paquete de datos hasta su destino

y deciden por cuál enviarlo. Además, están continuamente analizando el tráfico de la red y calculando estadísticas, por lo que tienen información real sobre la situación del mismo: conocen no sólo si una ruta está caída o no, sino también su grado de congestión. Igualmente, son capaces de "aprender" nuevas rutas. De hecho, mejoran el rendimiento cuando hay más de un camino posible para transmitir un mensaje, ya que encuentran la mejor ruta posible y la "aprenden". Sin ellos, el funcionamiento de Internet no sería posible.

Un *router* suele estar conectado a otros *routers*. Su funcionamiento es muy sencillo: cuando recibe un mensaje, elige a qué otro *router* enviarlo a continuación, empleando para ello todos los datos que posee acerca del tráfico de red (su decisión no es aleatoria), para que acabe llegando a su destino. Cuando el *router* sabe con cierta exactitud a qué ordenador va dirigido en último término el paquete de datos, entonces lo dirige o bien directamente a él o bien hacia la red en la que se encuentra dicho ordenador (y en ese caso lo puede recibir un *switch* o un *hub*), en lugar de enviarlo a otro *router*.

En algunas ocasiones, los *routers* son ordenadores dedicados exclusivamente a la gestión del tráfico de la red. En este caso, deben estar equipados con al menos dos tarjetas de red que soporten el protocolo IP y con un sistema operativo preparado a tal efecto. Por ejemplo, el sistema operativo Linux admite, entre sus opciones de compilación, la posibilidad de ser optimizado para funcionar sobre un equipo enrutador. Sin duda, el lector puede imaginar la potencia que puede lograrse con esta opción: en un ordenador se pueden usar potentes programas capaces de procesar múltiples datos y calcular diversas estadísticas, y también realizar un análisis exhaustivo de los datagramas que circulan, pudiéndose programar filtros potentes que permitan prohibir o regular a una cierta velocidad (ancho de banda) determinados tipos de paquetes.

Para que el lector pueda hacerse una idea del grado de sofisticación que puede alcanzar un *router*, mencionamos a continuación algunas de las características que incorporan estos dispositivos:

- Seguridad: desde el punto de vista de la seguridad de las comunicaciones, suele ser posible programar un *router* para que sólo los equipos autorizados puedan conectarse a él. También es posible programar un *cortafuegos*, programa específicamente dedicado a la protección de los puertos a través de los cuales se reciben los paquetes de datos.
- Filtrado: podemos programar el *router* para que determinados equipos conectados a él (o, mejor, a las redes que él gestiona directamente) sólo puedan enviar cierto tipo de paquetes.
- Administración: muchos *routers* incorporan herramientas de gestión que permiten controlar diferentes aspectos del funcionamiento de la red: altas o bajas de máquinas conectadas a la misma, velocidad (ancho de banda) variable en función de la subred, del tráfico o del tipo de paquetes, etc.

Pasarelas

En el campo de las redes de ordenadores, la palabra *pasarela* –es comúnmente empleada su voz inglesa *gateway*– es un termino genérico que puede referirse a tres tipos diferentes de dispositivos:

1.- Puede hacer referencia a un enrutador. De hecho, en muchas ocasiones, cuando se configura manualmente la conexión de un ordenador a una red existente (como, por ejemplo, la de la UJI) y el programa de configuración pregunta por la dirección IP de la pasarela o *gateway* de la red, en realidad está inquiriendo la dirección IP del dispositivo que hace de *router* para la misma.

2.- También puede referirse a una aplicación que se encarga de "traducir" los datos que procesa otro programa de aplicación de red. Por ejemplo, un *gateway* de correo electrónico, o de mensajes, convierte mensajes entre dos diferentes protocolos de correo electrónico.

3.- En su tercera acepción, tal vez la que se utiliza con mayor propiedad, hace referencia a un dispositivo, con frecuencia un ordenador, que realiza la conversión de protocolos entre diferentes tipos de redes (y, por tanto, traduce la información de un tipo de protocolo a otro).

En esta sección, nos estamos refiriendo precisamente a esta tercera acepción del término pasarela: un dispositivo que permite interconectar redes de arquitecturas muy diferentes y que usan protocolos de comunicación completamente distintos, mediante la conversión de los mismos. La traducción de las unidades de información contribuye a reducir la velocidad de transmisión a través de estos equipos. En realidad, podemos verlos como puertas de acceso, donde tiene lugar una conversión completa de protocolos.

Generalmente, la conexión desde una red local con otras redes en Internet requiere de una máquina que actúe como pasarela. Podemos decir, pues, que la función primordial de una pasarela es conectar redes locales (LAN) a Internet.

5.5. COMPONENTES *SOFTWARE* DE LAS REDES

Ahora que ya hemos hablado de los soportes o medios físicos que se emplean para la comunicación entre ordenadores y de los dispositivos, *hardware* en general, que sirven para habilitarla, es hora de que pasemos a hablar de los diferentes programas y protocolos que permiten la transmisión de los datos en realidad y el acceso a los diferentes servicios proporcionados: el *software* propiamente dicho. En este sentido, vamos a considerar:

- El programa que permite controlar y manipular la componente electrónica del adaptador empleado (módem, tarjeta de red...). Esto es lo que comúnmente conocemos con el nombre de *driver*, y que es necesario instalar para que el sistema operativo de nuestro PC pueda manipular el adaptador y hacer factible la conexión a una red de ordenadores existente.

- El protocolo de comunicaciones, es decir, el conjunto de "reglas" que hay que implementar en los programas que se encargan de gestionar el envío y la recepción de los datos con independencia del adaptador y del soporte empleados en último término.

El lector atento observará que la utilización de determinados soportes físicos que hemos comentado con anterioridad implicaban la utilización de ciertos protocolos a bajo nivel. En este caso no nos estamos refiriendo a ellos –y aunque en cierta forma lo sean, no los hemos considerado explícitamente como *software*– sino a los protocolos estándar que implementan los mecanismos básicos que permiten enviar y recibir datos sin pérdida, sin importar el medio físico y el adaptador empleados, y que constituyen el núcleo básico de las redes de ordenadores y de Internet. Normalmente, forman parte del sistema operativo de nuestro ordenador (constituyen módulos o bibliotecas del mismo).

- Los programas que permiten que los ordenadores conectados a una red puedan ofrecer determinados servicios, tales como el de páginas web, transferencia remota de ficheros, correo electrónico, etcétera. Todos ellos conforman lo que denominamos *software* del servidor, y que podríamos clasificar en:
  - Programas para la administración de la red como, por ejemplo, la asignación dinámica de direcciones IP (posteriormente veremos esto con más detalle), gestión de dominios, etc.
  - Programas de aplicación y servicios, como los que se dedican a servir páginas web, gestionar el envío y la recepción del correo electrónico, etc.
- Los programas que se utilizan en los ordenadores que, conectados a la red, actúan como "clientes" de los servicios ofrecidos. Son aplicaciones como los navegadores (clientes de los servicios de páginas web), gestores del correo electrónico (permiten gestionar los mensajes de una o varias cuentas de correo electrónico y, por tanto, son clientes de este servicio), programas para la transferencia de ficheros, etc.

5.5.1. Controlador del adaptador de red

Los adaptadores empleados con mayor frecuencia para la conexión a una red de ordenadores a través de un determinado medio físico son dos: los módems y las tarjetas de red (Ethernet). En el "saco" de los primeros vamos a considerar aparatos diversos, tales como los módems ADSL y los cable módems, cuyo mecanismo de funcionamiento es diferente.

En cualquier caso, suponemos que el lector es consciente de que cada tarjeta o módem, cada adaptador en general, es diseñado por un fabricante. El fabricante proporcionará un programa controlador o *driver* para este adaptador, ya sea directa (en un CD-ROM por ejemplo) o indirectamente (el fabricante publica las especificaciones de funcionamiento de su dispositivo para que otros, los programadores del kernel de Linux por ejemplo, desarrollen el controlador).

El *driver* suele ser, por tanto, un programa (o conjunto de programas) específico(s) que permite(n) al ordenador "dialogar" con la tarjeta de red o módem para que ésta entienda las órdenes que da el sistema operativo. De hecho, es el programa que usa el sistema para controlar el adaptador correspondiente (tarjeta o módem). Como ya hemos dicho, puede ya formar parte del propio sistema operativo o ser proporcionado por el fabricante.

De todas formas, se ha de ser consciente de que si nuestro sistema operativo no lo lleva incorporado o el fabricante no lo proporciona para nuestro sistema entonces no será posible hacerlo funcionar y no podremos conectarnos a la red.

Esto lo decimos porque algunos módems no tienen *drivers* disponibles para Linux, por lo que no podremos conectarnos a Internet con dicho sistema a no ser que cambiemos de adaptador.

La Figura 5.26 ilustra el proceso de selección y configuración del *driver* para una tarjeta Ethernet en un Windows 2000.

Si nos centramos en las tarjetas de red o tarjetas Ethernet, quizá el adaptador más popular para la conexión a una red de ordenadores, tenemos que:

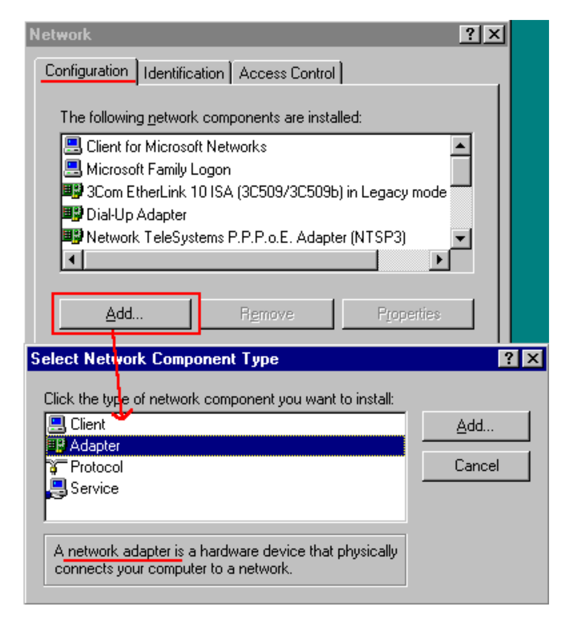


Figura 5.26. Selección y configuración del controlador del adaptador de red, en este caso una tarjeta Ethernet, en un sistema Windows

- En los sistemas operativos de la familia Windows, los controladores pueden venir ya incorporados, especialmente en el caso de tarjetas muy populares y no muy recientes, e instalarse automáticamente al detectarse la tarjeta. Por otra parte, es usual que el fabricante incluya el *driver* –con un programa de instalación preparado en CD-ROM– junto con el adaptador.

- En el caso de Linux, si el fabricante no proporciona un *driver* es posible que no podamos hacer funcionar la tarjeta durante un tiempo (hasta que alguien lo desarrolle). No obstante, hay que decir que en la actualidad prácticamente todas las tarjetas están soportadas de serie en Linux. En cualquier caso, antes de comprar una tarjeta de red que haya de funcionar bajo Linux, hemos de asegurarnos de que exista un controlador para ella.

En general, podemos decir que el *driver* se encarga de la gestión “de bajo nivel” de la conexión. Para que el lector pueda entender un poquito mejor lo que queremos decir con esto, pondremos como ejemplo el funcionamiento del *driver* de una tarjeta de red conectada a un red con topología en bus.

En nuestro ejemplo, cuando se desea efectuar una transmisión de información, el controlador de la tarjeta del emisor envía un paquete de datos al cable emitiendo inicialmente el número del receptor. Como el cable es compartido por todos los equipos conectados a la red, los ordenadores (o mejor dicho, los *drivers* de sus tarjetas) lo están monitorizando continuamente. Cuando uno de ellos detecta una transmisión dirigida a la tarjeta que está manipulando, el controlador se encarga de leer los datos y pasarlos al sistema operativo para su ulterior proceso.

Además, el *driver* se encarga de gestionar y resolver algunos de los problemas que ocurren frecuentemente en las comunicaciones. En nuestro ejemplo de la topología en

bus, es posible que dos o más ordenadores intenten emitir al mismo tiempo, provocando una *colisión*. De resolver estas dificultades, entre otras, se encarga el controlador.

5.5.2. Protocolos de red: TCP/IP

Una vez que se tiene una red preparada desde el punto de vista físico (nos estamos refiriendo al medio físico: tipo de cable, ondas de radio, etc.) es necesario definir cómo se va a utilizar este soporte de red para transmitir los datos. El conjunto de reglas que definen esa utilización constituyen los protocolos que se emplearán.

Los protocolos son, pues, los convenios que permiten "entenderse" a los computadores interconectados independientemente de la topología y de los medios físicos empleados. Es algo así como especificar el "idioma" en el que van a hablar. Hay varios protocolos que se utilizan en Internet, aunque también se utilizan cada vez más en redes locales no necesariamente conectadas a Internet, es decir, lo que se ha dado en llamar *intranets*:

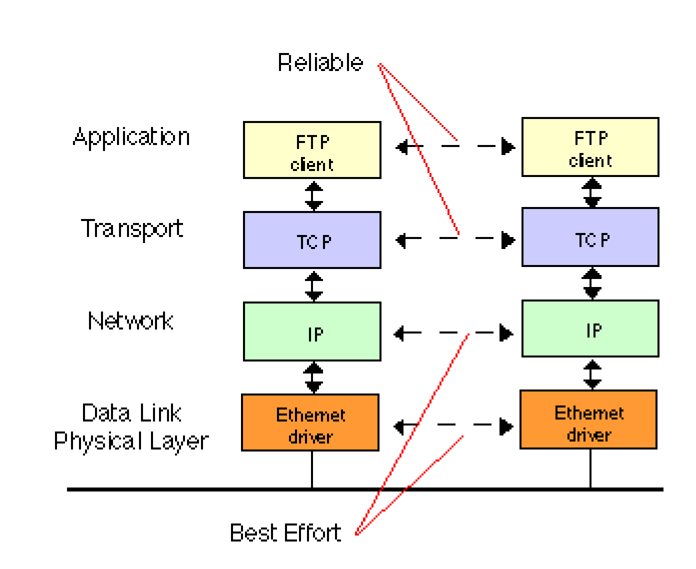


Figura 5.27. Diferentes niveles para la comunicación entre ordenadores basada en el protocolo TCP/IP

- Novell IPX,
- Apple Talk,
- Netbios/SMB (Windows),
- LAN Manager,
- X.25, y
- **TCP/IP**

Este último es, sin duda, el más importante de todos, ya que es el protocolo estándar para las comunicaciones en Internet. Por tanto, para poder conectar un ordenador a Internet su sistema operativo ha de disponer de un módulo que implemente el protocolo TCP/IP, que es el "lenguaje" que usan todas las redes interconectadas en Internet. En la actualidad, todos los sistemas operativos del mercado incluyen este módulo. Debe tenerse en cuenta que, a la hora de utilizar los diferentes servicios que proporciona Internet, se articula un mecanismo de "capas" que engloba los diferentes protocolos utilizados, tal como ilustra la Figura 5.27.

Internet Protocol (IP)

El protocolo IP es el protocolo en el que se basa Internet. Es un acrónimo del inglés *Internet Protocol*. El objetivo que persigue IP es proporcionar un sistema de comunicación que sea independiente de la topología (forma) de la red y de qué dispositivos físicos se empleen para establecerla. De esta manera, siempre se puede utilizar el protocolo IP, ya sea en una red tipo Ethernet –donde los ordenadores van equipados con tarjetas de red– o en una conexión empleando un módem ADSL, que a bajo nivel puede utilizar algún tipo de protocolo *punto a punto* (PPP, de *Point-to-Point Protocol*).

El funcionamiento de las comunicaciones en Internet se basa en dos principios fundamentales: simplicidad y reparto del trabajo en varios niveles (recuérdese la Figura 5.27). Cada protocolo se encarga de una (o algunas) tarea(s) concreta(s) con el objetivo de realizarla(s) a la perfección[23]. El protocolo IP se encarga del *encaminamiento*, esto es, de enviar los datos hasta su destino (es algo así como el cartero que se encarga de llevar las cartas a los respectivos domicilios).

Mediante IP, la comunicación siempre se basa en transmitir la información en forma de bloques de un determinado tamaño, paquetes de datos o *datagramas*, que viajan de una máquina a otra pasando por las redes y máquinas intermedias necesarias (a través de *routers*, pasarelas, etc.) y que siguen un formato específico y públicamente conocido.

Para hacer referencia e identificar correctamente a las máquinas a las que se quiere poder enviar datagramas –y que, por consiguiente, estarán conectadas a Internet– es necesario darles una dirección, de la misma forma que nuestro domicilio está identificado por una dirección y código postal para que funcione el servicio de correos y el cartero no se equivoque.

Es por ello que cada ordenador conectado a Internet tiene un número o *dirección IP* que lo identifica de forma única. De hecho, las direcciones IP son números de 32 bits que identifican tanto a la máquina como a la red a la que ésta pertenece –algo necesario para que los *routers* puedan hacer eficientemente su trabajo. Se distribuyen de modo que no pueda haber dos máquinas con el mismo número IP, aunque una misma máquina puede tener distintos números IP (por ejemplo, el ordenador portátil de uno de los autores, ya que tiene dos tarjetas de red, una de ellas inalámbrica, o bien por utilizar algún tipo de asignación dinámica, como el servicio DHCP que veremos posteriormente). La unicidad la garantiza un organismo internacional encargado de asignar estas direcciones a nivel mundial: la Autoridad de Números Asignados en Internet, IANA (*Internet Assigned Numbers Authority*), como vimos en la sección 5.2.

La comunicación que efectúa IP se produce mediante el envío de paquetes de datos (datagramas) en los que figura tanto la dirección IP del destinatario como la del remitente. La Figura 5.28 muestra el formato que tienen que tener los datagramas que transmite IP. Los datagramas se envían a enrutadores (*routers*) que escogen, de forma distribuida, una ruta entre emisor y receptor entre las varias posibilidades existentes.

Normalmente, los números o direcciones IP no se representan en binario, sino como una serie de cuatro números comprendidos entre 0 y 255. Es decir, se separan los 32 bits en cuatro grupos de ocho y se representa cada grupo mediante su equivalente en decimal. Para distinguir cada uno de los grupos, se suele separar cada grupo del siguiente

[23]Ya se sabe, cuando mucho se quiere abarcar es más fácil cometer errores.

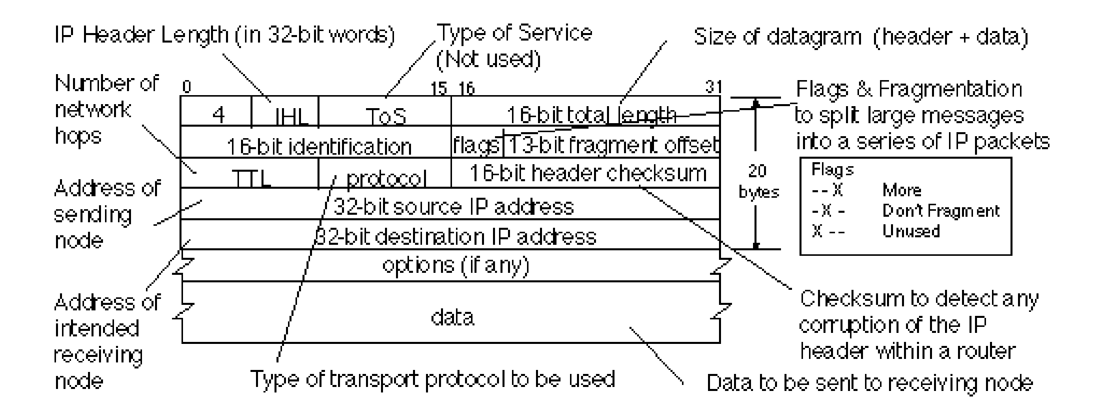


Figura 5.28. Formato específico que deben tener todos los datagramas enviados con IP

mediante un punto (`.`). Por ejemplo, el servidor de alumnos de la UJI, `anubis`, tiene como dirección IP `150.128.98.54`. Como ya hemos comentado anteriormente, cabe considerar cada dirección IP como si realmente estuviera dividida en dos partes: la primera identifica la red (el dominio, como veremos posteriormente) y la segunda el número que tiene la máquina dentro de ella. Esto es necesario ya que contribuye a una mayor eficiencia en el funcionamiento de la red, ya que así los enrutadores pueden ofrecer un mayor rendimiento.

Es importante conocer que el número de bits reservados en la dirección IP para identificar a la (sub)red no es fijo. Este número, además, influye (obviamente) en el número de bits que se emplean para identificar a las máquinas en la (sub)red (el resto). Ello tiene una influencia directa en el número de máquinas que pueden conectarse a una determinada red (cuantos más bits dispongamos para identificar ordenadores, más podremos conectar). Así, en función del número de bits dedicados a especificar la red y la máquina, podemos distinguir entre cinco clases de redes, aunque, en realidad, sólo tres de ellas tienen importancia en la actualidad desde nuestro punto de vista, algo menos "profesional". Se describe brevemente cada una de estas redes en el Cuadro 5.2.

Clase	Bits para la red	Máquinas en la red	Rango de direcciones
A	8	16.777.216	1.0.0.0 a 127.255.255.255
B	16	65.536	128.0.0.0 a 191.255.255.255
C	24	254	192.0.0.0 a 223.255.255.255
D	Especial para enviar a varios ordenadores		
E	Reservada para usos futuros		

Cuadro 5.2. Taxonomía de las redes en función del número de bits dedicados a identificarlas

Las redes que nos interesan son las de las clases A, B y C, que son las de uso corriente hoy en día. Las redes de los tipos D y E son especiales y están reservadas para uso futuro. Además, hay que tener en cuenta que existen ciertas direcciones y rangos *especiales* que deben tenerse en cuenta a la hora de administrar una red:

- El rango de direcciones comprendido entre 127.0.0.0 y 127.255.255.255, y las direcciones 0.0.0.0 y 255.255.255.255 tienen usos especiales y no pueden emplearse para asignar números IP arbitrariamente.

- Además, se han reservado, dentro de las distintas clases de redes, los rangos siguientes: de 10.0.0.0 a 10.255.255.255 para la clase A, para la clase B de 172.16.0.0 a 172.31.255.255, y de 192.168.0.0 a 192.168.255.255 para la clase C. Estas direcciones permiten hacer pruebas o bien instalar redes locales sin depender de nadie en el exterior, es decir, que se trata de montar redes locales cuyos ordenadores no van a estar *directamente* conectados a Internet (aunque pueden tener acceso indirecto a través de un *router* que sí que tiene asignada una dirección IP "normal"). Estas direcciones IP se conocen con el nombre de *direcciones IP privadas*, ya que son particulares de cada red y no importa que existan duplicados[24], ya que no existe el peligro de confusión al no ser posible su empleo para la conexión a Internet. Dependerá del administrador concreto de cada red cómo va a llegar el tráfico de Internet hasta cada uno de los equipos que usen estas direcciones (si es que es necesario que llegue). Esto, por supuesto, no influye a la hora de gestionar el tráfico "local" de la red: éste llegará a cada equipo empleando su dirección IP privada, por lo que es necesario que *no existan duplicados dentro de la propia red local* (no es posible tener dos ordenadores en la *misma* red local que usen la misma dirección IP, aunque ésta sea privada).

Como ya sabemos, los números o direcciones IP que sí que se van a emplear para la conexión directa de ordenadores a Internet no pueden asignarse arbitrariamente. Existe un organismo oficial internacional, la IANA, al cual deben solicitarse las direcciones IP que van a emplearse con este fin. Este tipo de direcciones, por contraposición, se conocen con el nombre de *direcciones IP públicas*, ya que van a ser públicamente conocidas (y utilizadas) para la conexión directa de equipos a Internet. Realmente, cada administrador de un dominio (red local o corporativa) recibe un rango de direcciones IP solicitadas a la IANA que debe administrar correctamente, encargándose de evitar duplicados de direcciones IP y de la correcta configuración de los equipos que se conectan a la red que gestiona. Si un administrador realiza una gestión incorrecta o negligente de su red se arriesga a recibir severas sanciones por parte de la IANA.

Protocolos de "acompañamiento"

Si bien IP es el protocolo básico de comunicaciones en Internet, que constituye su esencia y conforma su esqueleto, existen otros protocolos de "acompañamiento" que se utilizan en el mismo nivel o capa de comunicaciones que IP y que permiten gestionar sucesos a ese nivel o averiguar el número IP que corresponde a una máquina a partir de la dirección Ethernet (las que identifican a las tarjetas de red y que fueron explicadas en la sección 5.3.3 al introducir la topología en bus) o viceversa:
continúa en la página siguiente ...

[24] Probablemente la dirección IP 192.168.1.100 esté siendo utilizada en este momento por miles de ordenadores en el mundo, ninguno de ellos conectado directamente a Internet.

. . . viene de la página anterior

ICMP (acrónimo de *Internet Control Message Protocol*): se utiliza para informar de cosas que suceden en la red, como por ejemplo que los mensajes no pueden llegar a una máquina porque han excedido el tiempo permitido para su envío.

ARP (acrónimo de *Address Resolution Protocol*): se utiliza para transformar las direcciones IP en direcciones Ethernet.

RARP (acrónimo de *Reverse Address Resolution Protocol*): se utiliza para obtener la dirección IP a partir de la dirección Ethernet. Es útil para ordenadores que arrancan desde la red.

BOOTP: es parecido al anterior (RARP) ya que permite arrancar máquinas desde la red, pero con más flexibilidad.

En la actualidad, existe un problema grave en Internet que tiene que ver con sus "dimensiones" a escala global (léase número de equipos conectados) y que se debe, fundamentalmente, a una falta de previsión en la concepción del protocolo IP. Aunque el número de direcciones disponibles con la versión actual de IP (IPv4) es bastante grande (direcciones de 32 bits implican un número total de 2^{32} –4.294.967.296– direcciones distintas), la gran expansión de Internet hará que dentro de poco este número se quede corto.

Existen algunos "trucos" que sirven para paliar en parte este problema: algunas direcciones IP se asignan "dinámicamente" (bajo demanda), ya que se otorgan a un computador sólo cuando éste lo solicita (vía DHCP, por ejemplo). También puede ocurrir que varios equipos compartan una misma conexión a Internet a través de una única dirección IP pública mediante la utilización de direcciones IP privadas (que ya sabemos que se pueden repetir sin problemas). Los proveedores de Internet, que son empresas que ofrecen conexión pública a Internet, utilizan estas técnicas pues no todos sus usuarios están conectados simultáneamente.

En su momento, nadie imaginó que Internet podría interconectar a tantas máquinas. Pero, además, se necesita una mayor flexibilidad para poder tratar adecuadamente la conexión con dispositivos móviles y dar mayor seguridad a las comunicaciones.

Para poder solucionar satisfactoriamente (en principio) estos problemas se ha definido una nueva versión del protocolo IP, la versión 6, popularmente conocida como IPv6. Entre otras características, destaca que las nuevas direcciones IP tendrán 16 bytes (128 bits, lo cual permite un total de 2^{128} direcciones distintas), con lo que se tendrán direcciones suficientes para hacer frente en la práctica a cualquier crecimiento de Internet.

El protocolo IPv6 se irá implantando progresivamente en los próximos años y, de hecho, la mayoría de los sistemas operativos (Windows, Linux. . .) y muchos de los dispositivos (*routers*, *switches*. . .) que se comercializan actualmente ya vienen preparados para soportarlo.

Como ya sabemos, el protocolo IP usa los denominados paquetes de datos o *datagramas* para el envío de información en las comunicaciones. Los datagramas constituyen, pues, las unidades básicas de información que contienen datos y que permiten comunicarse en una red de conmutación de paquetes como es Internet. El término "conmutación de paquetes" quiere decir, básicamente, que no todos los paquetes necesitan

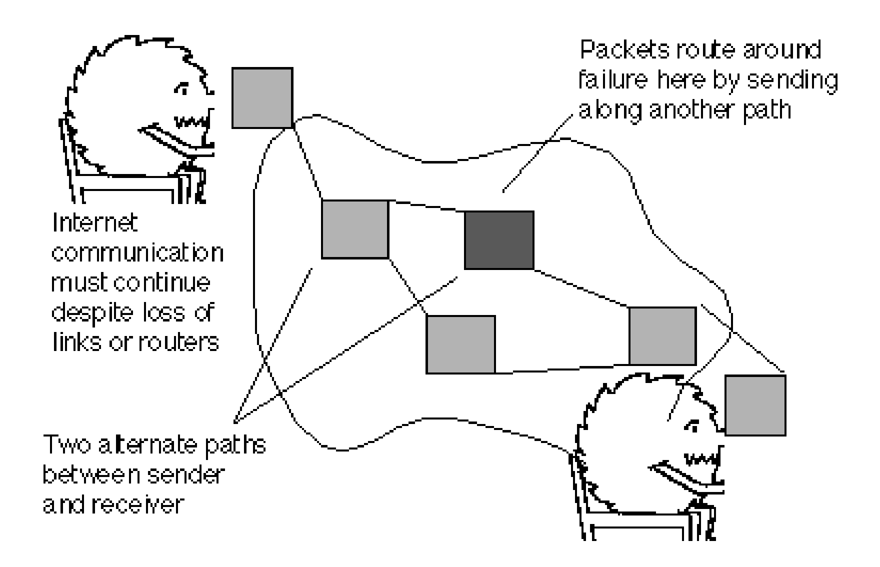


Figura 5.29. La comunicación en Internet debe continuar pese a la posible caída de enlaces o rutas. Los datagramas IP se envían por rutas alternativas hasta llegar a su destino

seguir la misma ruta hasta su destino, ni llegar al mismo tiempo o en un orden establecido. De esta manera, si un nodo, enlace o ruta "cae" o se satura, la comunicación puede continuar a través de otros caminos alternativos, tal como ilustra la Figura 5.29.

Transmission Control Protocol (TCP)

El protocolo TCP (acrónimo de *Transmission Control Protocol*) es quien se encarga de gestionar el control de las transferencias completas de datos a un nivel (capa) superior al del IP. TCP está destinado a proporcionar conexiones entre máquinas de manera fiable, incluso si la red por la que se transmiten los datos no lo es.

En general, esto es necesario ya que las redes *no son fiables* debido a que, aunque de forma mayoritariamente imperceptible por nosotros[25], se producen errores en las transmisiones por diferentes causas: ruido, cortes, averías en nodos, interferencias, etc. Por tanto, podemos afirmar que TCP es el protocolo que permite garantizar conexiones fiables en redes no fiables.

A diferencia de IP, las conexiones que proporciona TCP no utilizan el concepto de datagrama o paquete de datos, si bien TCP también utiliza una cierta estructura de datos para encapsular los "pedazos" de información que debe transmitir, la cual podemos apreciar en la Figura 5.30. Podemos ver los datos que transmite TCP como si fuesen una "corriente" (*stream*) de bits, de tal modo que se puede transmitir cualquier cantidad de información como un todo. En realidad, TCP está un nivel por encima del protocolo IP y, de hecho, se encarga de trocear la información y manejar a IP para que envíe los datagramas hasta su destino.

De esta manera, TCP se encarga de dividir los datos que hay que transmitir desde la máquina origen (un fichero, por ejemplo) en las partes necesarias, pasar estas partes a IP con los datos requeridos para que se puede rellenar la información de control de los datagramas (cabeceras) y que éste las pueda enviar a la máquina destino y, en

[25]Gracias, entre otras cosas, al protocolo TCP.

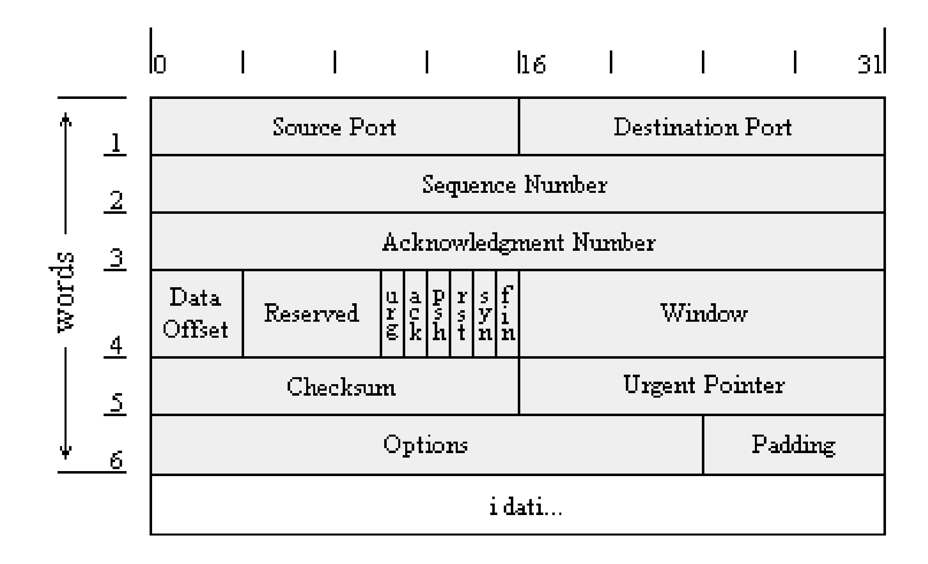


Figura 5.30. Estructura de datos que utiliza el protocolo TCP para transferir partes de información de una máquina a otra

ésta, reunir las partes para recuperar la información original (fichero completo). En la máquina receptora, pues, TCP junta los paquetes que le envía el protocolo IP, y lo hace en el *orden preciso* –aunque lleguen desordenados TCP se encarga de restaurar el orden original empleando la información de control que se rellenó en origen– para poder restituir el mensaje original.

El lector curioso se preguntará sin duda qué clase de información de control se envía para que TCP pueda restaurar sin problemas los datos originales. Esto queda fuera del alcance de esta asignatura, eminentemente introductoria a la materia, pero podemos decir que cada paquete de datos es completado con información tal como la dirección IP del ordenador remitente, bits de control (*checksum*) que permiten detectar errores de transmisión, la dirección IP del ordenador destino, etc.

Si el destino es ajeno a la red local, lo cual puede determinarse fácilmente examinando la dirección IP del destinatario[26], el paquete se envía a un ordenador o dispositivo privilegiado (normalmente un enrutador o *router*) que lo envía a la red destino –a través, posiblemente, de otros ordenadores que se encargan del enrutado. En la máquina destino, TCP, aparte de recuperar la información, se encarga igualmente de detectar si llegaron correctamente –y en un tiempo prudencial, *timeout*– todos los paquetes que la conforman. Si alguno no hubiese llegado, TCP solicita su reenvío a la máquina origen.

Es importante conocer que TCP permite establecer varias conexiones *simultáneas* por ordenador. Para ello, se emplea un número identificador de 16 bits que permite crear hasta 65.536 (que es el valor de 2^{16} puesto que se usan 16 bits) *puertos* virtuales diferentes. Un puerto TCP es, *grosso modo*, un punto de conexión en una máquina. Algunos de los puertos tienen un uso predeterminado por convenio (llamémosle estándar)

[26]Recordemos que la dirección IP no sólo permite identificar a la máquina, sino también a la red a la que ésta pertenece.

y ofrecen determinados servicios, como por ejemplo:

- El puerto TCP número 80, que se utiliza por defecto para todas las transferencias que tienen que ver con el servicio WWW (*World Wide Web*).
- El puerto TCP número 25, que se emplea para la gestión del servicio de correo electrónico.
- El puerto TCP número 21, que se usa para las transferencias de datos realizadas por el servicio FTP (transferencia de ficheros; *File Transfer Protocol*).

El modelo que sigue el TCP para efectuar las transmisiones de datos es, aproximadamente, el que describimos a continuación. Primero se establece una conexión desde la máquina origen hasta la máquina destino. Las conexiones se hacen contra un *puerto* de la máquina destino para, de esta forma, permitir más de una conexión por máquina. Una vez establecida la conexión en el puerto correspondiente, se transmite la información deseada. Finalmente, se cierra la conexión.

Como ya hemos comentado, los puertos no son más que números de dieciséis bits para identificar las conexiones. Los números bajos (menores que 256) corresponden a lo que se denominan *puertos bien conocidos* y sirven para servicios estándar, como los que hemos comentado anteriormente.

UDP (*User Datagram Protocol*)

El uso del protocolo UDP (acrónimo del inglés *User Datagram Protocol*) supone una alternativa al TCP.

Este protocolo lo utilizan algunas aplicaciones que quieren mandar datos de forma directa a un servidor sin necesidad de establecer previamente una conexión (por ejemplo, para realizar consultas al servicio DNS que estudiaremos posteriormente).

El protocolo UDP, básicamente, permite enviar un paquete IP a un puerto determinado de una máquina.

5.5.3. Servicios de Internet. *Software* del servidor

Sabemos ya que Internet es una red mundial de redes de ordenadores conectadas entre sí. Ahora bien, desde nuestro punto de vista como usuarios, cabe destacar el hecho de que esta comunicación entre los ordenadores nos permite disfrutar de una serie de servicios impensables hace unas pocas décadas: acceder remotamente a otro ordenador, enviar o recibir ficheros, enviar y recibir mensajes de correo electrónico, consultar información en páginas web (WWW), compartir ficheros, etc.

Todos estos servicios han convertido a Internet en una herramienta fundamental con la que se puede acceder a una cantidad ingente de información y establecer comunicaciones con otros usuarios. Hoy en día, y gracias al mecanismo de puertos del protocolo TCP, cualquier máquina conectada a Internet puede ofrecer cuantos servicios desee y el resto de ordenadores acceder a ellos. Esto es lo que se conoce popularmen-

te con el nombre de *modelo cliente-servidor*, paradigma de funcionamiento de la red mundial de ordenadores.

Todas las redes que se interconectan para constituir la red mundial Internet funcionan según el modelo (o arquitectura) *cliente-servidor*. Esto significa que ciertos ordenadores en la red actúan como *servidores* ya que *ofrecen* una serie de servicios, mientras que otros ordenadores actúan como *clientes* ya que *acceden* a dichos servicios. Hay que tener en cuenta que estos roles no son fijos, ya que un mismo ordenador puede actuar a veces como cliente o a veces como servidor, e incluso adoptar ambos papeles simultáneamente.

Un ejemplo de servicio podría ser el de páginas web, popularmente conocido como WWW (por *World Wide Web*). Para poder ser cliente de este servicio, necesitamos instalar en nuestro ordenador un determinado programa que actúe como tal. En este caso, se trata de instalar un navegador de Internet, tal como *Firefox* o *Internet Explorer*.

En general, lo más habitual es que tanto servidores como clientes utilicen programas específicos para comunicarse con otros ordenadores de la red. El servidor necesita un programa que *atienda las peticiones* para un servicio determinado (y probablemente en un puerto o puertos TCP establecidos previamente). El cliente necesita un programa que *efectúe las peticiones* por nosotros de manera adecuada. En el ejemplo anterior, una máquina que desee servir páginas web podría utilizar el programa *Apache* a tal fin, mientras que el programa (navegador) *Firefox* podría ser el programa que usase otro ordenador para poder visualizar esta páginas.

Dentro de Internet es posible encontrar una gran cantidad de servicios. Entre los más comunes tenemos (aunque bien podríamos decir que se trata tan sólo de una "punta de lanza", dada la gran cantidad de servicios existentes):

- Servicio de resolución de nombres: DNS.
- Asignación automática de dirección IP en un rango determinado: DHCP.
- Servicio de conexión remota: telnet.
- Servicio de transferencia remota de ficheros: FTP.
- Servicio seguro de conexión y transferencia de ficheros remotas mediante técnicas criptográficas: ssh y scp.
- Correo electrónico.
- Acceso a paneles de noticias: news.
- Compartición de ficheros entre iguales: P2P (Napster, Kazaa, e-Donkey, etc.).
- Compartición remota de unidades (discos o particiones): NFS, SMB.
- Acceso a salas de charla (*chat*): IRC.
- Servicio de páginas web: World Wide Web.

En las posteriores secciones comentaremos algunas cosas sobre el funcionamiento de los servicios más populares y, precisamente, veremos cuál es el significado de las siglas que hemos empleado en la lista anterior. El lector debe tener presente cuando abordemos estos servicios que, según el modelo cliente-servidor, para acceder a ellos es necesario tener un "soporte" adecuado en el ordenador servidor –es decir, un

programa servidor adecuadamente configurado– y utilizar el programa pertinente –en el sentido de que permita que nuestro ordenador se "comunique" con el programa servidor– en el ordenador cliente.

El *software del servidor*, que es, en general, cualquier ordenador que proporciona un servicio en Internet, se encarga, fundamentalmente, de atender las peticiones de los clientes en el puerto o puertos TCP correspondientes y generar las respuestas adecuadas para ellos. Estos programas específicos, necesarios para que una máquina pueda ofrecer servicios en una red, pueden formar parte en algunas ocasiones del propio sistema operativo o bien, lo más habitual, requerir una instalación aparte. En el ordenador cliente se deberán instalar los programas capaces de "dialogar" con los programas que prestan dichos servicios si se desea acceder a ellos. Brevemente, procedemos a exponer algunos de los programas que permiten ofrecer los servicios más conocidos:

- Servicio de resolución de nombres (DNS): *bind9*.
- Servicio de páginas web (WWW): *Apache*, *Microsoft Internet Server*…
- Servicio de compartición remota de unidades (discos duros, particiones, impresoras, etc.): *samba*, *nfs*.
- Servicio de transferencia remota de ficheros (FTP): *wu-ftpd*, *proftpd*…
- Servicio seguro de conexión y transferencia remotas con criptografía: *openSSL* y *openSSH*.
- Correo electrónico: *sendmail*, *exim*, *postfix*, *qmail*…
- Bases de datos corporativas (con capacidad de acceso remoto): *MySQL*, *PostgreSQL*, *Oracle*.

***Domain Name System* (DNS)**

DNS, acrónimo del inglés *Domain Name System*, es un *servicio* de Internet que está estrechamente relacionado con la utilización, por "debajo", de direcciones IP. Ya sabemos que, tal como está estructurado el protocolo IP, para comunicarse con cualquier máquina es necesario dar su dirección, que es un número de 32 bits. Ahora bien, ¿es necesario tener que memorizar los números IP de todas aquellas máquinas conectadas a la red con las que deseamos establecer una comunicación?

Hace algo más de tres décadas, al poco de nacer Internet (más concretamente, su predecesora ARPANET y las primeras redes que se le adhirieron), la respuesta a esta pregunta era que sí. No obstante, las direcciones IP son números difíciles de recordar, resulta bastante incómodo tener que hacerlo y, además, ocurre que ciertas máquinas pueden cambiar dinámicamente su número asignado por distintas circunstancias.

Precisamente para facilitar el manejo de las direcciones IP se creó el *sistema de nombres de dominio* (DNS), que es un servicio de Internet que permite establecer una asociación entre los denominados *nombres de dominio* (seguidamente abordaremos esta cuestión) y las direcciones IP de redes y computadores en Internet. Por ejemplo, el nombre de dominio del servidor de alumnos de nuestra Universidad, denominado `anubis` y que tiene la dirección IP `150.128.98.54`, es `anubis.uji.es`.

Un *dominio* es un ámbito de una red en el que puede haber varios ordenadores, cada uno con su dirección IP. Por ejemplo, la red de la UJI constituye un dominio en

Internet. De esta manera, se divide Internet en distintos dominios de alto nivel. A su vez, cada dominio puede dividirse en *subdominios* “incluidos” dentro de él. Algunos de los dominios de más alto nivel son genéricos (por “temas”): `com` (comercial), `gov` (gobierno de EE.UU.), `edu` (Universidades e instituciones educativas en general en EE.UU.), `mil` (fuerzas armadas de EE.UU.), `org` (organizaciones sin ánimo de lucro), `info` (para información, aunque éste es relativamente reciente), etcétera.

Sin embargo, otros dominios de nivel jerárquico superior agrupan los subdominios por países, derivándose de éstos. Así, tenemos el dominio `de` para Alemania, el dominio `es` para España, el dominio `jp` para Japón, el dominio `nl` para Holanda, el dominio `fr` para Francia, el dominio `ca` para Canadá, el dominio `ch` para Suiza (“ch” es por Confederación Helvética), etcétera.

Estructuralmente, Internet está formado por subredes más pequeñas, que, a su vez, están formadas por otras más pequeñas, y así sucesivamente, siguiendo una estructura jerárquica. La idea básica en la que se basa DNS consiste en representar las direcciones IP mediante nombres (cadenas, esto es, secuencias de caracteres) que se organizan de manera jerárquica siguiendo la estructura de dominios existente: son los denominados *nombres de dominio*. Para ello, los nombres que asociamos a una red separan cada subdominio del siguiente en la jerarquía mediante puntos y se “leen” de derecha a izquierda –de “mayor” a “menor” en la jerarquía.

Por ejemplo, `www.uji.es` es el nombre de dominio de un ordenador cuya dirección IP es `150.128.82.253` y que está formado por tres subdominios. Leyendo de derecha a izquierda tenemos, en primer lugar, el dominio de nivel jerárquico más alto: el dominio `es` (España) en el que se engloban (entre otros) todos los subdominios correspondientes a las redes de las universidades de nuestro país (y el de la UJI más concretamente). Después tenemos el dominio `uji`, que denomina a la red de la Universidad Jaume I, la cual está conectada, a su vez, con las otras redes de las universidades españolas (REDIRIS). Finalmente, tenemos el nombre `www`, que corresponde al servidor web de la Universitat Jaume I.

En realidad, el conjunto de todos los nombres de dominio se divide en distintas zonas que abarcan partes de los distintos dominios. Cada una de estas zonas tiene un servidor de nombres primario y, posiblemente, uno o más servidores secundarios, que se encargan de devolver la correspondiente dirección IP a todos aquellos clientes que preguntan por un nombre determinado. De esta manera, cuando una máquina necesita conocer la dirección IP asociada a un nombre dado pregunta a alguno de los servidores que conoce: si alguno sabe la respuesta se la da, en caso contrario cada uno de ellos busca el nombre en los servidores conocidos en su nivel jerárquico superior, los cuales a su vez responderán si la conocen y, si no, trasladarán la pregunta a otros servidores de nombres. Eventualmente, la pregunta puede no trasladarse a un nivel jerárquico superior, sino a servidores inferiores si se sabe que un determinado dominio por el que se pregunta es, en realidad, un subdominio dentro del nivel al que pertenece el servidor de nombres interrogado.

Vamos a intentar abordar con mayor precisión el funcionamiento del servicio DNS. Lo primero que hemos de tener en cuenta es que al configurar la conexión de cada ordenador que se conecta a Internet se asigna uno o más servidores de nombres, que, normalmente, son máquinas de la propia red local a la que conectamos el ordenador en cuestión.

Cuando se solicita establecer una conexión mediante cualesquiera de los servicios disponibles a una máquina empleando su nombre (lo más común) –por ejemplo a `www.google.com` mediante un navegador WWW– el sistema operativo del ordenador recurre, de forma transparente para el usuario[27], al (a los) servidor(es) de nombres que se ha(n) asignado en el paso anterior.

Si éste (alguno de éstos) sabe qué dirección IP se corresponde al nombre por el que se pregunta, se responde en el acto. Si no lo sabe(n), el (los) propio(s) servidor(es) recurre(n) a otro(s) predeterminado(s). El proceso consiste en seguir la propia jerarquía de dominios: si nadie en el dominio (subred) propio sabe la respuesta, se acude al nivel superior (la red que comprende a ésta) o inferior (alguna de las subredes comprendidas), según convenga. Finalmente, el servidor de nombres obtiene la respuesta y resuelve el problema planteado por el cliente. A todo este proceso, que transcurre en un tiempo normalmente imperceptible para el ser humano, se le denomina *resolución de nombres*. Este procedimiento se ilustra de manera gráfica en la Figura 5.31.

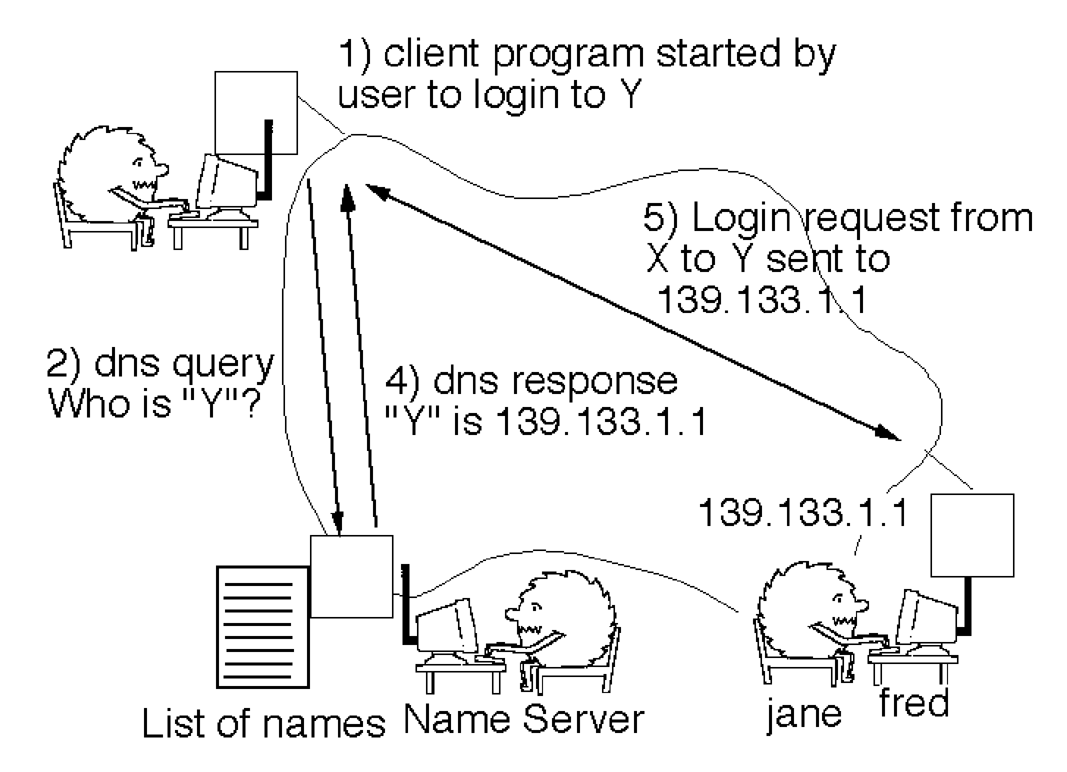


Figura 5.31. Proceso de funcionamiento del servicio DNS. Fred, que está trabajando en el ordenador X, quiere establecer una videoconferencia con Jane, que está trabajando con el ordenador Y, empleando el servicio adecuado. El ordenador X lanza una petición al servidor de nombres (DNS) que tiene asignado preguntando por la dirección IP de la máquina Y. Éste consulta en su lista de nombres (eventualmente, si no lo encuentra, puede efectuar consultas a otro servidor DNS) y cuando averigua la dirección IP, se la pasa de vuelta a X (quien inició la petición). En ese momento, desde el ordenador X ya se puede establecer la conexión con la máquina Y empleando su dirección IP (`139.133.1.1`)

Ahora bien, ¿cómo identificamos desde nuestro ordenador a los servidores de nombres que se nos han asignado? No puede ser mediante sus nombres porque, en ese caso,

[27]Es decir, sin que el usuario se dé cuenta.

¿quién resuelve los nombres de los servidores de nombres?[28] Parece un trabalenguas, pero, en realidad, esto plantea un pequeño problema. Realmente no es tal problema ya que la solución es muy sencilla: usaremos la dirección IP para identificar a los servidores de nombres. Así, nuestro ordenador puede ponerse directamente en contacto con ellos. Esto no supone tampoco molestia alguna para nosotros: los administradores de la red en la que nos encontramos nos proporcionan estos datos a la hora de configurar la conexión a la misma, o bien han configurado algún tipo de servicio (DHCP, por ejemplo) que proporciona estos datos (y otros) de forma automática a nuestro ordenador en el momento de establecer la conexión a la red.

El sistema de nombres de dominio o DNS está *distribuido* en la propia Internet: no hay una base de datos única que asocia direcciones IP con nombres de dominio. Aunque hay definidos una serie de servidores de nombres principales (en el nivel jerárquico superior) a nivel mundial, cada subdominio es responsable de gestionar sus propios servidores de nombres y "enlazarlos" con los de los niveles jerárquicos superior e inferior.

Por ejemplo, hay un par de máquinas de la UJI que resuelven todos los nombres de ordenadores y dominios de la propia Universidad. Cuando estos servidores no saben resolver un nombre, pasan a consultar a algún servidor a nivel superior (posiblemente en Madrid). Supongamos que se consulta por un nombre correspondiente a un dominio español, tal como `www.uam.es`. El servidor a nivel jerárquico superior a los de la UJI conoce la dirección del dominio `uam.es`, así que pide a un servidor de nombres de dicho dominio que le diga la dirección de `www.uam.es` para notificársela al que hizo la petición desde la UJI.

Supongamos que, por contra, se trata de un nombre correspondiente a un dominio extranjero, tal como `www.google.com`. El ordenador español que atiende la petición de los servidores de la UJI a nivel jerárquico superior puede consultar a otro ordenador "por encima" de él para que averigüe la dirección.

Por último, hay que tener presente que la asociación entre direcciones IP y nombres de dominio no tiene por qué ser necesariamente única. Es posible definir varios nombres de dominio asociados a una misma dirección IP, mediante mecanismos de creación de *alias*.

Una analogía para describir el funcionamiento de Internet

Un modelo de funcionamiento similar a Internet es el del *servicio postal*. El programa cliente recibe los datos que queremos enviar y se los pasa al TCP, que es algo así como la central local de correos.

TCP divide la información en trozos y se los pasa al IP, que es algo así como el cartero. IP añade una serie de información adicional: dirección IP del remitente y dirección IP del destinatario, bits de control –para que la máquina destino pueda juzgar si, por problemas físicos de la red, el mensaje se ha deteriorado–, etc., y envía el paquete de datos a la "oficina de correos" (puntos de salida de la red local), la cual la transferirá a otra y así sucesivamente hasta llegar al destinatario.

continúa en la página siguiente . . .

[28]O, planteado de otra forma, ¿qué fue antes, la gallina o el huevo?

. . . viene de la página anterior

En destino, IP recibe los paquetes, comprueba la dirección del remitente y chequea la integridad de la información recibida (haciendo uso de los bits de control). Si la información es correcta, pasa el paquete a "su" TCP.

El TCP del destinatario recibe la secuencia de paquetes, la ordena correctamente y reconstruye la información original, pasándola al nivel superior, esto es, al programa que está esperando recibir esos datos.

Aunque las tecnologías sean diferentes, el funcionamiento del servicio postal es muy similar al de Internet.

Las diferentes partes o redes que forman Internet se conectan entre sí a través de un conjunto de dispositivos denominados *routers* (encaminadores o enrutadores).

Los *routers* son como subestaciones postales que toman decisiones sobre la elección de la ruta por la cual van a circular los paquetes de datos.

Cada *router* no tiene por qué tener una conexión directa con cada uno de los demás *routers* de la red.

Siguiendo el símil del servicio postal: la oficina de correos local envía el sobre a otra oficina, ésta a otra y así hasta llegar a su destino final.

Esto significa que cada *router* sólo necesita conocer qué conexiones están disponibles y cuál es el mejor *destino intermedio* para llevar el paquete más cerca del destinatario.

El *router* lee, por tanto, la dirección a la cual va dirigido el paquete de datos y decide dónde va a enviarlo (cuál es el mejor camino).

Dynamic Host Configuration Protocol (DHCP)

El servicio DHCP (acrónimo del inglés *Dynamic Host Configuration Protocol*) es un servicio relativamente reciente en Internet (sobre todo si lo comparamos con servicios más tradicionales como FTP o el correo electrónico). Este servicio se encarga de proporcionar, de *manera automática*, los parámetros necesarios para la configuración de la conexión de un ordenador a una red, sin que el usuario tenga que proporcionar ningún dato –tan sólo es necesario que el ordenador tenga permiso para acceder a este servicio. DHCP consta de dos "partes" fundamentalmente:

- En primer lugar, DHCP constituye un *protocolo de comunicaciones* que le permite suministrar parámetros específicos de configuración desde un servidor DHCP hacia la máquina cliente, tales como la dirección IP de la máquina que hace de pasarela o *gateway* en la red local y cuáles son las direcciones IP de los servidores de nombres (DNS). Es necesario proporcionar todos estos datos (y algunos más) para configurar correctamente la conexión de cualquier equipo a la red.

  Antes, esos datos eran proporcionados en papel por los administradores (con el consiguiente riesgo de equivocación por parte de usuarios legos en la materia), o bien eran ellos los que tenían que establecer personalmente esta configuración (con la consabida molestia). Con DHCP los administradores configuran los parámetros una sola vez (en el servidor DHCP), ya que éste se encarga de transmitir automáticamente estos datos a los equipos conectados a la red. Tan sólo

es necesario instalar un programa cliente del servicio DHCP en cada ordenador que se desee conectar y configurarlo para iniciarlo cada vez que se arranque el ordenador. La mayoría de sistemas operativos usados (por ejemplo, Linux y Windows) incluyen clientes DHCP "de serie".

- En segundo lugar, DHCP puede verse como un *mecanismo para asignar direcciones IP* a los ordenadores clientes del servicio (los que se conectan a la red) dentro de un rango determinado a priori por los administradores. De esta manera se consigue asignar direcciones IP únicas, de forma automática y sin miedo a equivocaciones (tal como duplicar una dirección IP por error). La asignación de direcciones IP por parte del servidor DHCP puede ser:

    - **Automática**: de esta manera, el servidor DHCP es configurado para asignar una dirección IP de forma *permanente* a un ordenador cliente del servicio, que puede ser identificado por el "número" del módem empleado para la conexión, el identificador de la conexión propiamente dicho, o la dirección de su tarjeta Ethernet (en el caso de una red de este tipo). La identificación es necesaria para comprobar la autorización de acceso al servicio.
    - **Dinámica**: con este sistema, el servidor DHCP asigna una dirección IP al ordenador cliente sólo *por un periodo de tiempo determinado*. Esta forma es la única que permite *reutilizar* aquellas direcciones IP del rango que no están en uso momentáneamente. Es útil para asignar números IP a los clientes que se conectan temporalmente (como por ejemplo las conexiones caseras) o bien para compartir un rango limitado de direcciones IP, ya que facilita su reutilización cuando no están en uso.
    - **Manual**: en este caso, es el administrador de la red quien se encarga de asignar las direcciones IP a los clientes de *manera fija* (almacenando la asignación en una tabla o en un fichero) y DHCP meramente se limita a transmitirlas según el identificador del cliente.

En la actualidad, DHCP es el sistema que emplean mayoritariamente los proveedores de Internet para asignar las direcciones IP y los datos de configuración de la conexión a sus clientes en el momento de establecer la conexión a Internet desde sus casas. Normalmente, los proveedores de Internet utilizan la asignación dinámica, ya que sus características encajan con el "perfil" de la mayoría de sus usuarios (conexiones y desconexiones diarias en franjas horarias concretas, lo cual implica que se produzcan varias liberaciones de direcciones IP asignadas). No obstante, hay algunos que todavía emplean la asignación automática y otros que, por un coste adicional para el cliente, usan asignación manual.

Progresivamente, el servicio DHCP se ha venido utilizando cada vez más por parte de los administradores de redes locales de cierta envergadura, tales como las de las universidades y grandes y medianas empresas, por su comodidad tanto para el administrador como para el usuario.

***World Wide Web* (WWW)**

El servicio de páginas web, también conocido como WWW (acrónimo del inglés *World Wide Web*), es el servicio que más ha contribuido a revolucionar y popularizar Internet. WWW ha disparado de forma dramática el crecimiento y uso de Internet, hasta el punto de que mucha gente lo confunde con la propia Internet. Obviamente no son lo mismo, ya que Internet es la red mundial de redes de ordenadores y WWW constituye un servicio más dentro de la misma.

World Wide Web es un servicio que comenzó a desarrollarse en el año 1989, y que fue ideado por Sir Tim Berners-Lee en el CERN (acrónimo del francés *Conseil Européen pour la Recherche Nucléaire*) de Ginebra. El CERN es lo que conocemos en España como la Organización Europea para la Investigación Nuclear.

Básicamente, lo que ocurrió fue que Berners-Lee, ante la necesidad de distribuir e intercambiar información acerca de sus investigaciones de una manera más efectiva, desarrolló, junto con su grupo de investigación, el lenguaje HTML (acrónimo del inglés *HyperText Markup Language*) o lenguaje de etiquetas de hipertexto; el protocolo de comunicación HTTP (acrónimo del inglés *HyperText Transfer Protocol*), que constituye el núcleo del servicio WWW; y el sistema de localización de objetos en la web denominado URL (por el inglés *Universal Resource Locator*), que describe las reglas para la escritura de las direcciones web (y que veremos posteriormente).

WWW, por tanto, es un servicio que permite acceder a ficheros que contienen *hipertexto*. El hipertexto es un documento que contiene texto, imágenes y referencias a otros ficheros (posiblemente también conteniendo hipertexto). La idea es tener páginas de texto que tienen enlaces (*hipertexto e hiperenlaces*) a otras páginas con información relacionada. Estas páginas se escriben en el lenguaje especial HTML, y se transmiten mediante el protocolo de comunicación HTTP.

Poco a poco, el lenguaje HTML se fue extendiendo para permitir la inclusión de gráficos y distintos efectos (tablas, mapas, formularios. . .). El efecto es la creación de una gran red de información vinculada mediante enlaces. World Wide Web significa literalmente “telaraña de cobertura mundial”. Hoy en día, es posible encontrar prácticamente cualquier información sobre los más diversos temas en “la web”, así como acceder a ficheros de gran tamaño. Para acceder a dicha información hay programas cliente especializados en el acceso al servicio WWW: son los navegadores (en inglés, *browsers*, es decir, “hojeadores”).

Para “acelerar” las descargas de grandes ficheros existen programas denominados *download managers* o gestores de descargas. Estos programas permiten, entre otras cosas, continuar la transferencia si ésta se interrumpe desde el punto en que se cortó (lo que se denomina *resume download*). Se puede decir que WWW es la aplicación más popular de Internet (quizá solo equiparable al correo electrónico) y, como hemos dicho, para mucha gente Internet es sinónimo de WWW.

Desde el punto de vista del servidor, debe haber un programa que se encargue de atender (“escuchar”) y de gestionar las respuestas a las peticiones (mensajes) que llegan al puerto TCP asignado al servicio web. Este puerto es, por defecto, el puerto 80 como vimos cuando describimos el mecanismo de puertos TCP.

Las peticiones y respuestas siguen el convenio establecido por el protocolo de comunicaciones HTTP y se escriben en código ASCII (o alguna de sus extensiones en la

actualidad), que es texto plano, por lo que son directamente legibles por cualquier ser humano. Por tanto, si alguien utiliza un programa de escucha en la red –programas que se encargan de monitorizar todo el tráfico de paquetes que circulan por la red y que se denominan popularmente *sniffers*– tiene acceso a toda la información que se envía durante la comunicación entre nuestro navegador y el servidor web.

Esto plantea un grave problema de seguridad, ya que, por lo visto, el protocolo HTTP no es seguro *per se*. En consecuencia, debemos tener mucho cuidado tanto a la hora de recibir como a la hora de transmitir la información que consideremos confidencial. Precisamente, para poder garantizar la confidencialidad en las comunicaciones que lo requieran, como por ejemplo las transacciones electrónicas que hacemos con nuestro banco a través de la web, se utiliza el protocolo denominado HTTPS (HTTP seguro o *secure HTTP*), que no es más que el protocolo HTTP tradicional con una capa SSL que añade cifrado a las comunicaciones. SSL (acrónimo del inglés *Secure Socket Layer*) proporciona un nivel de conexiones seguras que incorporan técnicas criptográficas de cifrado de datos.

El tercer elemento desarrollado por Berners-Lee y su grupo de investigación consiste en una especificación de las reglas que el navegador debe obedecer para identificar recursos en la web. Esto es lo que llamamos URL (*Universal Resource Locator* o localizador universal de recursos, en castellano). Los navegadores, a instancias de los usuarios, solicitan URL al servidor web siguiendo el convenio establecido para nombrar los recursos accesibles vía WWW. Veamos algunos ejemplos que nos servirán para ilustrar la cuestión:

- `http://www.unawebopedia.com/index.html`
- `http://www.unawebopedia.com:80/index.html`
- `http://www.unawebopedia.com/cgi-bin/programa?saluda=1&nombre=Pepe\Paz`

De esta manera, podemos apreciar que un URL presenta una determinada estructura que se compone de varios elementos:

(a) El primer elemento consiste en la especificación del *protocolo de comunicaciones* que va a establecer el navegador con un determinado servidor. El protocolo por defecto (si no se especifica ninguno) es `http`, lo cual implica, como ya sabemos, establecer una comunicación con un servidor web. Sin embargo, los navegadores modernos entienden de varios protocolos de comunicación para diferentes servicios, como por ejemplo `https` (también contra un servidor web pero configurado con SSL), `ftp` (acceso a un servicio FTP, que veremos a continuación), `news` (para el acceso a un servidor de noticias, que son como “tablones de anuncios” en Internet) o `file` (para el acceso a ficheros en el disco duro local).

(b) Después, viene el *nombre* o *dirección IP* de un servidor. Ya sabemos que si usamos el nombre (lo más habitual), el sistema operativo de nuestro ordenador arrancará previamente una consulta al servidor (o servidores) DNS correspondiente para averiguar el número IP y efectuar la comunicación, entonces, con el servidor web. Si empleamos la dirección IP, entonces la comunicación se establecerá de forma directa, sin necesidad de realizar una consulta previa al servicio DNS.

(c) Opcionalmente, y separado por dos puntos (:) del nombre o dirección IP, se puede indicar el *puerto* TCP al cual irá dirigida nuestra solicitud. Por defecto, se utilizará el puerto establecido según el protocolo empleado. Si estamos usando HTTP (lo más corriente si estamos utilizando un navegador) este puerto es el 80. En algunas ocasiones, es posible que los administradores de un servidor web lo hayan configurado para atender las peticiones en otro puerto distinto; en ese caso, será necesario indicar el número del puerto que se ha utilizado cuando queramos conectarnos con un navegador.

(d) A continuación, y separado por una barra (/), tenemos la *ruta* que hay que seguir en el disco del servidor para acceder a un determinado *recurso*. En general, hablamos de "recurso" para hacer referencia tanto a un fichero que contiene un cierto tipo de datos (podría ser, por ejemplo, una página web, o sea, un fichero de texto escrito con el lenguaje HTML) como al resultado de ejecutar un programa que ofrece ese servidor. En el primer caso (fichero) hablamos de *contenido estático*, mientras que para el segundo (programa) decimos que se trata de *contenido generado dinámicamente*.

(e) Finalmente, y sólo en el caso de que el recurso al que accedemos sea un programa, puede seguir una *relación de parámetros con sus valores correspondientes* (esto ya dependerá de las características del programa en cuestión). El comienzo de la relación de parámetros se separa de la ruta al programa mediante un interrogante (?) y cada parámetro se separa entre sí del siguiente mediante el carácter &. La relación de parámetros tiene la forma "nombre del parámetro=valor".

Como ya sabemos, desde el punto de vista del cliente hemos de emplear un navegador para acceder al servicio WWW. Los navegadores más utilizados son, por este orden, *Internet Explorer* de Microsoft y *Mozilla Firefox*. El primero es usado de forma mayoritaria, pero el segundo es el navegador que más cuota de mercado ha ganado en los últimos años. Otros navegadores conocidos son *Opera*, *Safari* y *Galeón*.

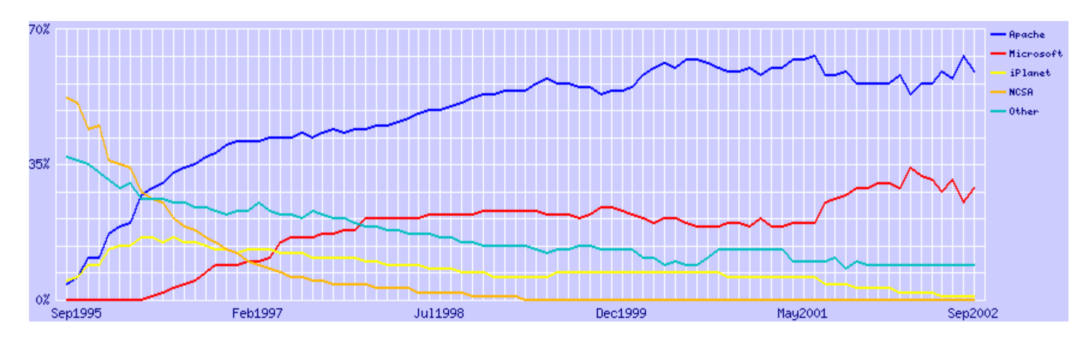


Figura 5.32. Evolución de la cuota de mercado de varios programas para la gestión del servicio web entre 1995 y 2002. *Apache* con una cuota alrededor del 60 % se sitúa claramente en cabeza, seguido por *Internet Server* de Microsoft con poco más de un 30 %. En la actualidad, las diferencias entre ambos han disminuido, ya que *Internet Server* ha ampliado su cuota de mercado a costa de *Apache*

También es necesario usar un programa que gestione el servicio en el ordenador servidor. Aquí ya es más probable que el lector conozca menos programas. En la actualidad, hay varios servidores de páginas web, tales como *Apache* y *Microsoft Internet*

Server, los dos grandes dominadores del mercado mundial. *Apache* ha sido siempre el programa más dominante (es *software* libre, gratuito, modular y multiplataforma) y es el más popular: tiene muchos módulos que implementan diversas extensiones y funcionalidades extras. Pero su clara hegemonía se está viendo amenazada recientemente por *Internet Server*, que ha experimentado un fuerte auge en los últimos tiempos. La Figura 5.32 muestra la evolución de la cuota de mercado de los programas para servidores web en el periodo comprendido entre septiembre de 1995 y septiembre de 2002.

File Transfer Protocol (FTP)

FTP (siglas de *File Transfer Protocol*) es un protocolo de comunicaciones que implementa un servicio que permite la transferencia de ficheros por Internet. Este servicio (y consecuentemente el protocolo) es muy antiguo en Internet (fue uno de los primeros servicios que se ofrecieron) y, por tanto, es muy fiable. La idea básica consiste en abrir una sesión FTP con otra máquina (el servidor FTP), ofreciéndose entonces la posibilidad de "bajar" o descargar y de "subir" o dejar ficheros (menos frecuentemente) desde/hacia una parte del sistema de ficheros del servidor.

Para establecer la sesión es necesario emplear el programa cliente adecuado, darle la dirección IP o el nombre del ordenador con el que deseamos conectarnos y un nombre de usuario con la contraseña correspondiente. Ese nombre de usuario y contraseña deben estar registrados en el servidor.

Ahora bien, ciertos servidores FTP han sido configurados para permitir establecer *conexiones anónimas*, es decir, de cualquier persona. Para ello suelen emplear como nombre de usuario el de uno "especial", `anonymous` normalmente (a veces también se puede utilizar `ftp`), y como contraseña o bien nada, o bien una dirección de correo electrónico. El propósito de un servicio de FTP anónimo es el de poder dejar ficheros y documentos públicos (no confidenciales, obviamente) a disposición de cualquier persona con acceso a Internet. Para el resto de usuarios se exige disponer de una cuenta abierta en el servidor, como hemos comentado en el párrafo anterior.

Cuando se establece una conexión mediante un cliente FTP, el servidor obedece a órdenes escritas en código ASCII, o sea, texto plano legible por cualquier ser humano. De hecho, el protocolo FTP no usa cifrado y toda la información circula por la red en forma de texto, incluyendo el nombre de usuario y la contraseña. Por tanto, son válidas las mismas precauciones que comentamos para el servicio WWW mediante HTTP, ya que cualquiera puede tener acceso a estos datos confidenciales empleando un *sniffer* en la red. Existe otro servicio similar, denominado SFTP (*Secure FTP*), que utiliza cifrado de datos mediante una capa SSL, de forma análoga a HTTPS. En el caso de un FTP anónimo no hace falta emplear estas precauciones, ya que lo que se desea es acceso libre por cualquiera y los ficheros y documentos que allí se depositan no son confidenciales (¡o no debieran serlo!).

En la práctica, no hace falta aprender las órdenes a las que responde el servidor FTP, salvo para casos puntuales donde se requiere efectuar un acceso desde consola (como es el caso de las prácticas de nuestra asignatura), ya que existen clientes gráficos que permiten establecer conexiones con servidores FTP y gestionar a "golpe de ratón" (arrastrando y soltando) las transferencias de ficheros. Además, los navegadores web actuales soportan el protocolo FTP: basta con escribir `ftp://` en lugar de

`http://` al principio del URL. Si deseamos acceder con el navegador a un servicio de FTP anónimo, a continuación escribiríamos el nombre del servidor simplemente. Por contra, si deseamos realizar un acceso con usuario y contraseña, escribiríamos a continuación `usuario@nombre_del_servidor_FTP` y esperaríamos a que el navegador mostrase una ventana preguntándonos la contraseña de acceso. Veamos algunas de las órdenes a las que responde un servidor FTP (en el material de apoyo de la práctica 8 se amplían datos al respecto):

- `user <usuario>`: se utiliza para introducir la identificación del usuario con el nombre `<usuario>` y así poder establecer una sesión FTP con el servidor. Previamente se ha de establecer conexión con el servidor FTP remoto mediante la orden `open <nombre_del_servidor>`.
- `pwd`: se emplea para averiguar cuál es el directorio activo en el servidor FTP (el directorio activo es aquél en el que se van a ejecutar las órdenes de transferencia o listado de ficheros, por ejemplo).
- `cd <directorio>`: realiza el cambio al directorio `<directorio>` –pasando éste a ser ahora el directorio activo– en el sistema de ficheros del servidor FTP.
- `get <fichero>`: para descargar el fichero `<fichero>` desde el directorio activo del servidor FTP a nuestro disco duro.
- `put <fichero>`: para dejar ("subir") el fichero `<fichero>` en el directorio activo del servidor FTP desde nuestro disco duro.
- `quit` o `bye`: cierra la conexión con el servidor FTP y finaliza la ejecución del programa cliente.

Para ilustrar el uso de este servicio, veamos un ejemplo de una sesión FTP establecida por el usuario `amarzal` en el servidor `nuvol.uji.es` empleando el programa cliente `ftp` ejecutado desde una consola de Linux (en modo texto, ya que `ftp` no es un programa gráfico):

(I) Primero, el usuario se conecta con el servidor `nuvol.uji.es` y se identifica introduciendo su contraseña.

```
amarzal@linux:~/infbas> ftp nuvol.uji.es
Connected to nuvol.uji.es.
220 nuvol.uji.es FTP server (Version wu-2.6.1-20) ready.
Name (nuvol.uji.es:amarzal):
331 Password required for amarzal.
Password:
230 User amarzal logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
```

(II) A continuación, el usuario solicita consultar el nombre de los ficheros que sirve el ordenador (listado de ficheros y directorios en el directorio activo):

```
ftp> ls
500 'EPSV': command not understood.
227 Entering Passive Mode (150,128,98,40,246,51)
150 Opening ASCII mode data connection for directory listing.
total 272
drwx------   2 822      521          4096 Sep 16 12:56 Mail
-rw-r-----   1 822      521           160 Apr 23  1996 Makefile
drwxr-x--x   2 822      521          4096 Jan 12  1998 News
-rw-r-----   1 822      521         21596 Mar 31  1996 adq.tcl
drwxr-x--x   2 822      521          4096 Feb 20  2001 bin
-rw-r-----   1 822      521            72 Apr 23  1996 bucle
-rw-r-----   1 822      521           209 Feb  4  1996 err
drwx------   2 822      521          4096 Jan 18  1999 mail
drwx------   2 822      521          4096 Sep 29  1998 nsmail
226 Transfer complete.
```

(III) El usuario solicita bajar una copia del fichero `adq.tcl` desde el directorio activo del servidor hasta el disco duro del ordenador desde el cual está ejecutando el programa cliente:

```
ftp> get adq.tcl
local: adq.tcl remote: adq.tcl
227 Entering Passive Mode (150,128,98,40,74,134)
150 Opening BINARY mode data connection for adq.tcl (21596 bytes).
100% |*************************************| 21596    48.43 KB/s  00:00 ETA
226 Transfer complete.
21596 bytes received in 00:00 (45.60 KB/s)
```

(IV) Finalmente, el usuario cierra la sesión FTP y finaliza la ejecución del cliente.

```
ftp> bye
221-You have transferred 21596 bytes in 1 files.
221-Total traffic for this session was 22384 bytes in 1 transfers.
221 Thank you for using the FTP service on nuvol.uji.es.
amarzal@linux:~/infbas>
```

Conexión y transferencia remotas y seguras (OpenSSH)

Existen servicios en Internet que ofrecen la posibilidad de efectuar conexiones remotas, es decir, abrir terminales desde otra máquina para ejecutar órdenes y programas en el servidor. Este tipo de servicios (concretamente el servicio Telnet) fue popularizado en su día por los servidores UNIX, aunque en la actualidad los ofrecen no sólo los servidores con sistemas operativos de la "familia" UNIX (como es el caso de Linux) sino otros, como por ejemplo ciertas versiones de Windows.

Como hemos dicho en el párrafo anterior, el pionero de este tipo de servicios fue Telnet. Telnet es un servicio de Internet, que fue muy utilizado durante varios años, y que permite a un usuario establecer una sesión de trabajo con un servidor al cual tiene acceso mediante un mecanismo de *conexión remota*. En una ventana de su pantalla el usuario puede escribir órdenes que ejecuta un ordenador que puede estar a miles de kilómetros de distancia. Para ello, es necesario que el sistema operativo del servidor al que nos conectamos permita la identificación de usuarios (es decir, que sea multiusuario) y que, por supuesto, el usuario tenga una cuenta en dicho servidor. La Figura 5.33 muestra un ejemplo de conexión remota al servidor `nuvol.uji.es` empleando el programa cliente `telnet`.

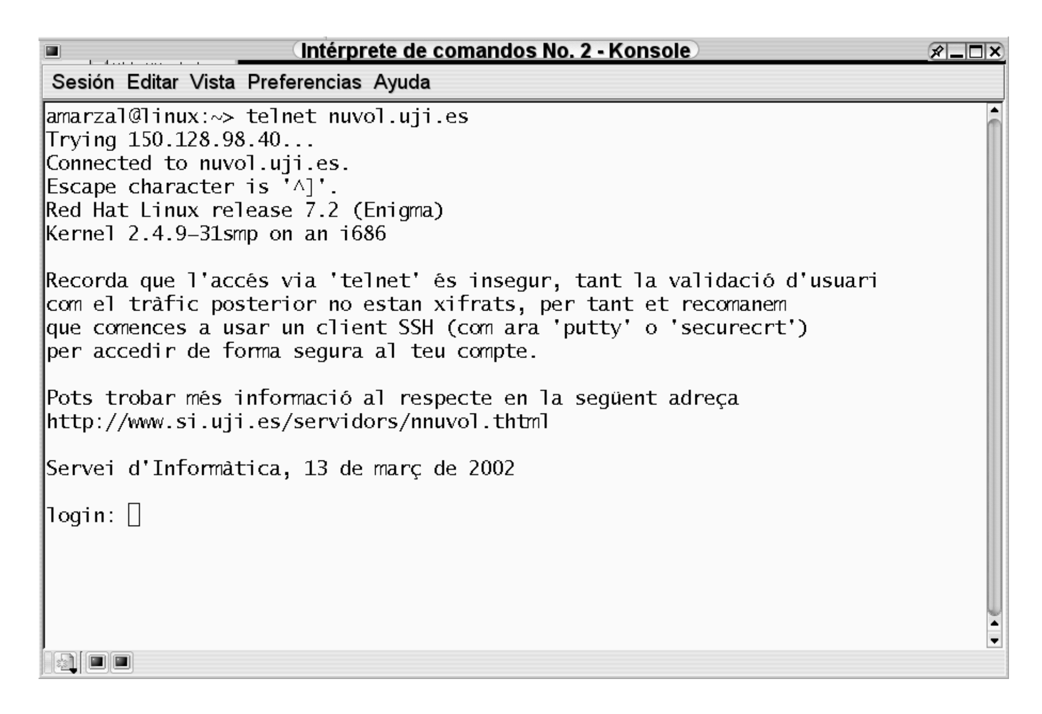


Figura 5.33. Utilización del servicio Telnet: conexión remota al servidor `nuvol.uji.es` mediante el programa cliente `telnet` ejecutado desde una consola de Linux

Para establecer la conexión, es necesario dar el nombre o la dirección IP del servidor al que deseamos conectarnos. Algunos programas cliente usan por defecto el usuario que hayamos utilizado en la máquina que empleamos para ejecutar el cliente y realizar la conexión. Si se quiere cambiarlo o bien si el servidor no permite identificar usuarios, habrá que especificarlo mediante las opciones adecuadas del programa cliente. En cualquier caso, el servidor pedirá que introduzcamos la contraseña del usuario. En el momento en que nos conectemos, los programas y órdenes que ejecutemos se estarán ejecutando realmente en el servidor, aunque nosotros observemos el resultado en la pantalla de nuestro ordenador.

El problema grave que presenta este servicio es el de la seguridad. Al igual que los protocolos de comunicación FTP y HTTP, Telnet no cifra la información (ni siquiera el nombre de usuario y contraseña), la cual viaja en texto plano legible por cualquier usuario que disponga de un *sniffer* (un programa capaz de monitorizar el tráfico de la red, como ya dijimos anteriormente). Es por eso que cayó en desuso hace ya unos cuantos años y hoy en día incluso está deshabilitado su uso por los administradores de redes, pues supone un agujero de seguridad importante. También el servicio FTP hubiese caído en desuso de no ser por la posibilidad de conexión anónima, la cual es una manera fácil y cómoda de poner datos a disposición pública en la red.

Aunque ahora existe el servicio Telnet-SSL, que incluye una capa SSL para cifrar las comunicaciones, Telnet ha sido sustituido por herramientas OpenSSH, que igualmente utilizan tecnología SSL (criptografía) en la transmisión de datos. Las herramientas OpenSSH ofrecen alternativas a:

- Telnet: mediante la utilización del servicio SSH (acrónimo de *secure shell*), podemos obtener exactamente la misma funcionalidad de *conexión remota* que pro-

porciona el servicio Telnet (y algunas cosas más, pues SSH incluye características adicionales), pero con *seguridad* ya que se cifran todos los datos transmitidos. En consecuencia, recomendamos el uso de `ssh` como sustituto del `telnet` tradicional.

- FTP: el mismo servicio SSH permite, igualmente, realizar transferencias *seguras* de ficheros mediante el cifrado de la información transmitida. Para ello, podemos emplear un programa cliente de SCP (acrónimo de *secure copy*) que permite a un usuario identificado en el servidor (hay que tener cuenta) tanto descargar como dejar ficheros. También recomendamos el uso de `scp` como sustituto de `ftp`. Recordemos que otra herramienta que puede usarse en este caso es SFTP (FTP combinado con SSL), tal como vimos en la sección anterior.

El funcionamiento en ambos casos es similar: el programa cliente recibe los datos del usuario, los cifra y los envía al programa servidor que es capaz de descifrarlos y tratarlos adecuadamente. Los paquetes de datos viajan de forma segura ya que nadie es capaz de leer su contenido (a no ser que rompa la clave, lo cual es casi imposible si se hacen las cosas bien). SSH utiliza un puerto diferente al de Telnet y FTP ya que se consideran servicios distintos y, por tanto, pueden coexistir. Debes saber que, además, SSH (concretamente su capa SSL) puede utilizarse, en conjunción con otros programas, para ofrecer servicios que, en principio, no soportan el uso de criptografía.

Correo electrónico

El servicio de correo electrónico, popularmente conocido por el anglicismo *e-mail* (abreviatura de *electronic mail*) es un servicio muy antiguo de Internet (de los originales) que permite la transmisión de mensajes a usuarios desde unos ordenadores a otros. Inicialmente estaba pensado para mandar mensajes de texto de una persona a otra. Después se extendió el protocolo de modo que se pudieran enviar también documentos y que fuese fácil mandar mensajes a más de una persona (listas de correo) o hacer que el correo se dirigiese de una dirección a otra, por ejemplo, por estar de vacaciones.

El protocolo de comunicaciones que constituye la base del servicio de correo electrónico se denomina SMTP (siglas de *Simple Mail Transfer Protocol*) y usa el puerto TCP número 25 para la recepción y envío (transferencia) de mensajes. Éste es el protocolo que permite que enviemos a Internet nuestros mensajes de correo electrónico. Según este protocolo, cada usuario puede disponer de una o más direcciones de correo, las cuales tienen el siguiente aspecto: `usuario@subdominio.dominio` (`jcamen@uji.es`, por ejemplo).

El servicio de correo, en realidad, lo ofrece un programa que se encarga de transmitir el correo de máquina a máquina hasta llegar al destino, momento en el que deja una copia en un fichero determinado (normalmente, con el nombre del usuario correspondiente). Dicho programa debe implementar el protocolo SMTP, como hemos visto.

A la hora de gestionar el buzón de correo electrónico hay que tener en cuenta que, normalmente, los usuarios no se conectan físicamente a la máquina que recibe el correo, sino remotamente. Habitualmente, las empresas proveedoras de Internet y muchos administradores de redes permiten consultar el correo directamente en el servidor. Para ello, ofrecen páginas web que proporcionan una interfaz de consulta y edición de

correo: son *pasarelas* que permiten gestionar cómodamente el correo a través de navegadores web (como por ejemplo la pasarela Webmail de la UJI).

No obstante, la gestión del correo mediante la pasarela, *exige* que estemos conectados todo el tiempo (mientras leemos los mensajes, mientras redactamos respuestas o nuevos mensajes, etc.) cuando esto no es realmente necesario. Sólo hemos de estar conectados el tiempo necesario para *recoger* el correo y para *enviarlo*. Si se disfruta de una *tarifa plana* en la conexión a Internet desde el hogar, esto daría igual. Pero si lo que se tiene es una tarifa donde se paga en función del tiempo de conexión o el volumen de datos transferidos (como es el caso de la mayoría de opciones existentes para la conexión a Internet mediante la red de telefonía móvil 3G), resultaría más conveniente procesar el correo y contestar a los mensajes en la máquina local, sin necesidad de estar conectado.

Por todo esto, puede resultar conveniente en algunas ocasiones descargar el correo hasta nuestro ordenador y manipularlo localmente. En este caso, el servidor de correo electrónico debe utilizar, además, un servicio de *descarga de correo*. Los servicios más populares relacionados con la gestión remota del correo electrónico son[29]:

- POP (acrónimo inglés de *Post Office Protocol*) es un protocolo que permite traer los mensajes de correo electrónico desde el servidor hasta el ordenador local, almacenándolos en el disco duro y *borrándolos* del servidor. Toda la gestión del correo se hace, pues, en el ordenador local. Para configurarlo adecuadamente, hay que especificar la dirección IP o el nombre del servidor, el nombre de usuario y la clave del mismo en dicho servidor. Habitualmente opera en el puerto 110. Recientemente se definió una extensión que permite dejar los mensajes en el servidor, de manera similar a IMAP (que veremos abajo). La diferencia está en que, en este caso, se pueden "vincular" los mensajes del servidor a la copia local, de manera que cuando se borra un mensaje en el ordenador local, también se borrará en el servidor.

- IMAP (acrónimo inglés de *Internet Message Access Protocol*) es un protocolo que, como POP, también permite descargar el correo a un ordenador local. A diferencia de POP, lo que se transmite es una *copia* de los mensajes, quedando el correo original en el servidor. Igualmente, para configurarlo hay que especificar la dirección IP o nombre del servidor y la clave y nombre de usuario en el mismo. Por tanto, la gestión del correo se realiza en realidad en el servidor: si queremos borrar un mensaje tendremos que conectarnos al servidor y borrarlo de allí. Habitualmente opera en el puerto 143. Permite una serie de configuraciones adicionales con respecto a las que proporciona POP.

En principio, el protocolo SMTP original sólo permitía transmitir mensajes empleando los caracteres del código ASCII (7 bits). No obstante, existen distintas extensiones (MIME, *Multipurpose Internet Mail Extensions*) que permiten enviar mensajes que emplean otras codificaciones (ISO-8859-?, UTF-8, Unicode) y ficheros adjuntos (incluso binarios, es decir, no sólo de texto, sino imágenes o programas, por ejemplo).

El envío de correo electrónico por la red padece de ciertos problemas de seguridad: los mensajes pasan sin cifrar por gran cantidad de máquinas con lo que no se garantiza

[29]Se les podría considerar servicios complementarios al de correo electrónico.

ninguna privacidad. Para conseguir la privacidad deseada se pueden emplear técnicas de criptografía, como por ejemplo PGP, que veremos en el capítulo 7: “Seguridad y protección de la información”. PGP tiene la ventaja de que se puede integrar fácilmente con muchos programas dedicados a la gestión del correo electrónico.

News: grupos de noticias

El servicio de *news* (noticias, en castellano) fue uno de los servicios más populares en Internet hasta hace poco tiempo, aunque en la actualidad ha sido un poco desplazado por el servicio de mensajería instantánea y por los foros de usuarios que ofrecen varias páginas web y portales de Internet. Es un servicio parecido al correo electrónico (podemos descargar mensajes, editarlos, publicarlos, responder, etc.), pero donde los mensajes se publican para que sean leídos por muchas personas (directamente o previa aprobación de un moderador).

De hecho, el objetivo que se persigue al publicar un mensaje en un tablón de noticias es que cualquiera pueda leerlo. La idea básica consiste en crear lo que se llaman *grupos de discusión* –fundamentalmente para obtener información y ayuda u opinar al respecto– en torno a algún tema concreto de interés. En principio cualquier persona puede mandar mensajes al grupo, tanto para responder a mensajes anteriores como para iniciar nuevos “hilos” de discusión. En algunos casos los grupos están moderados. Entonces el mensaje se envía al moderador que, si lo considera oportuno, lo publica en el tablón correspondiente al grupo.

Existen infinidad de grupos de noticias. Para facilitar su consulta y manejo sus nombres se organizan jerárquicamente de forma parecida a lo que ocurre con las direcciones IP y los nombres de dominio en Internet (servicio DNS), aunque nada tienen que ver con ellos realmente. Algunas de las “jerarquías” de más alto nivel son: `comp` (temas relacionados con los computadores), `sci` (temas científicos), `news` (discusión sobre el funcionamiento del propio servicio de noticias, creación y gestión de grupos), `misc` (temas variados de todo tipo), etc.

Además existen jerarquías de grupos de noticias clasificadas por países, por ejemplo `es` agrupa a todos los grupos de noticias para España (en castellano), y también locales –por ejemplo, la UJI tiene su propia jerarquía `uji`. Mención aparte merece la jerarquía `alt`. Aquí podemos encontrar todo tipo de temáticas, lo que incluye varios grupos sobre religión o sexo que pueden resultar polémicos en algunos contextos. Veamos algunos ejemplos de grupos de noticias en Internet:

- `comp.lang.c`: son noticias sobre computadores y computación en general (`comp`), en el grupo de noticias sobre lenguajes de programación (`lang`), y más concretamente sobre el lenguaje de programación C (`c`).
- `es.rec.tv.concursos`: son noticias de España en castellano (`es`), sobre temas recreativos (`rec`), centradas en temas de televisión (`tv`), concretamente sobre `concursos`.
- `es.rec.tv.series`: son noticias de España en castellano (`es`), sobre temas recreativos (`rec`), centradas en temas de televisión (`tv`), concretamente sobre `series`.

- `alt.rec.camping`: son noticias sobre temas alternativos (`alt`), concretamente recreativos (`rec`), acerca de temas de acampadas (`camping`).
- `alt.sex.oral`: son noticias sobre temas alternativos (`alt`), centrados en cuestiones de sexo (`sex`), y más concretamente sobre la práctica, usos y hábitos del sexo oral (`oral`).

Los grupos de noticias pueden consultarse tanto a través de programas especializados, como a través de la web: `http://groups.google.com` y `http://webforum.uji.es` permiten una cómoda consulta de los mensajes en los tablones de los grupos de noticias mediante nuestro navegador web preferido.

Compartición de ficheros entre iguales (P2P)

El servicio de "compartición de ficheros entre iguales" o *peer-to-peer file sharing*, popularmente conocido como P2P, ha sido creado para que múltiples usuarios de todo el mundo puedan compartir sus ficheros (documentos, imágenes, vídeo, audio, música, etc.). Todos los usuarios que desean compartir sus ficheros se conectan a un mismo servidor especificando su nombre o dirección IP y, opcionalmente (pues hay un puerto por defecto), el puerto TCP donde el programa que gestiona el servicio en el servidor está escuchando. Realmente, los ficheros a compartir residen en las máquinas de los clientes (los usuarios que se conectan), sin pasar nunca por el servidor. En resumen, los clientes se conectan a uno o más servidores para:

- publicar ficheros y ponerlos a disposición de otras personas, y
- saber qué ficheros se han publicado en los ordenadores de los otros usuarios conectados.

En su origen, el servicio funcionaba de tal forma que, una vez se había consultado al servidor y se había elegido el fichero a descargar, se establecía una conexión particular (P2P: *peer-to-peer*) entre los dos ordenadores clientes para intercambiar el fichero, sin mayor concurso del servidor.

Básicamente, lo que proporcionaba el servidor eran mecanismos de búsqueda eficientes para localizar ficheros concretos e iniciar una transferencia directa entre los ordenadores clientes, esto es, el servidor se encargaba de "poner en contacto" a los usuarios, ya que conocía las direcciones IP de todos los ordenadores conectados. Para ello, el servidor guardaba, de forma adecuada, la lista de los nombres de los ficheros a compartir e información del usuario y la máquina donde estaban localizados.

En la actualidad, el funcionamiento de este servicio ha evolucionado. El concepto de "descarga compartida" implica que existen varios usuarios que poseen todo o partes del fichero (son las denominadas "fuentes") y que hay otros tantos que están descargando el fichero por partes (a trozos). Con ello, se consigue optimizar el ancho de banda de la red de pares *considerada globalmente*: es mucho más rápido efectuar, pongamos por caso, 100 descargas simultáneas de forma mancomunada (por partes) de un fichero de 1 GB que efectuar esas 100 descargas una a continuación de la otra[30]. Además, se

[30]Recuerda que el protocolo TCP/IP en el que se basa Internet es un sistema de conmutación de paquetes.

utiliza un sistema de créditos por el cual el servidor da más puntos a aquellos clientes con mayor ancho de banda y que permiten que se descargue desde ellos un mayor volumen de datos. En base a estos puntos o créditos, el servidor pone a los clientes en contacto con mayor o menor número de fuentes (a más créditos, más fuentes).

Así, por ejemplo, es como funciona la red *e-Donkey* (*e-mule*), pero hay otras variantes, como por ejemplo BitTorrent, que combinan las características del sistema de créditos antes mencionado con el esquema de descarga directa original del P2P. En BitTorrent el enlace al fichero que queremos descargar contiene también información acerca de su localización.

Uno de los principios que siguen estos servidores es garantizar la privacidad de los usuarios conectados en la medida de lo posible: los usuarios usan apodos y se enmascaran los nombres y direcciones reales de las máquinas conectadas. Sin embargo, es muy fácil averiguar las direcciones IP de los clientes que están compartiendo un determinado fichero para cualquiera que *esté conectado al servidor*.

Abordemos, ahora, algunas cuestiones legales por las que, tristemente, las redes P2P son famosas. Y es que este tipo de servicios se han utilizado principalmente para intercambiar contenidos audiovisuales y programas con licencia, lo que ha acarreado, acarrea y acarreará numerosos problemas legales por violaciones de *copyright* y derechos de autor; cuestiones relacionadas con la denominada propiedad intelectual en suma.

Tomemos como ejemplo el caso de *Napster*, compañía pionera en la utilización y difusión de este servicio, inicialmente restringido al intercambio de ficheros de audio en formato MP3. Napster, que empleó el servicio P2P con la idea del intercambio de ficheros de música (MP3), cerró al cancelarse su servicio por orden judicial, ya que se consideró que Napster se usaba, fundamentalmente, para vulnerar derechos de autor y propiedad intelectual.

No obstante, todavía hoy siguen existiendo multitud de servidores “no oficiales” de la antigua red Napster repartidos por todo el mundo. Además, permiten buscar otras cosas aparte de MP3. Otras redes P2P han recogido igualmente el testigo: OpenNapster, Kazaa, e-Donkey, BitTorrent... Ponemos, a continuación, unos párrafos extraídos del capítulo 7: “Seguridad y protección de la información”, que arrojan algo de luz sobre estos temas legales.

La posibilidad de obtener copias *de información digital virtualmente* indistinguibles *del original hace que sea atractivo copiar programas sin pagar por ellos, ya que el* coste *de la copia es, por regla general,* ridículo. *Y esto no es sólo aplicable a los programas de ordenador, si no que también es aplicable a cualquier medio audiovisual digital (películas en DVD, CD de audio, etc.). Las* copias no autorizadas son ilegales*; los programas están protegidos por la ley de la propiedad intelectual y la nueva reforma del código penal (Ley Orgánica 15/2003 en vigor desde el 1 de octubre de 2004) que, entre otras cosas, penaliza la* distribución pública, con ánimo de lucro *y en perjuicio de tercero, de material con copyright. En este sentido, la ley 23/2006 (julio 2006) modifica el texto refundido de la* Ley Orgánica *sobre* Propiedad Intelectual*: básicamente, amplía el canon a una mayor variedad de soportes digitales (reproductores MP3, por ejemplo), pero mantiene el derecho de* copia privada*; aunque en este caso se restringe a un ámbito “familiar” y se ha de usar un medio legal para efectuarla (con lo que el autor de estos apuntes entiende que se mantiene igual que hasta ahora, pues el P2P no*

es ilegal).

Los denominados “piratas” vulneran los derechos de autor copiando y distribuyendo (cuando no vendiendo) copias no autorizadas por el propietario. El autor tiene derecho a decidir si su obra puede copiarse y en qué condiciones. No obstante lo dicho en el párrafo anterior, conviene distinguir entre dos conceptos totalmente distintos:

- Copia de seguridad: referida exclusivamente a los programas de ordenador y juegos. No se necesita autorización del autor para hacerla, ni se puede impedir por contrato. Consiste en reproducir (copiar) software para la utilización legítima del mismo por parte del usuario. El concepto “legítimo” es importante: sólo puede hacer la copia de seguridad para su propia utilización el propietario legítimo del software.

- Copia privada: referida a obras (literarias, musicales, audiovisuales, etc.). Las obras ya divulgadas pueden reproducirse sin autorización del autor para uso privado del copista siempre que la copia no sea objeto de utilización colectiva ni lucrativa. No se requiere ser el propietario legítimo de la obra: de ahí que se pague, por ejemplo, el “famoso” canon de los CD y DVD a la SGAE. Si bien, el concepto de divulgación previa es relevante. Aunque resulta ambiguo, parece que el legislador hacía referencia a cosas como, por ejemplo, la divulgación de películas por canales públicos de televisión (y su copia en VHS).

Dicho esto, la conclusión del autor de estos apuntes (que coincide con la de varios reputados expertos en Derecho) es que la utilización de las redes P2P para la descarga de *obras* (películas, música, libros. . .) sin ánimo de lucro *no es ilegal*, pues no constituye delito al tratarse de copias privadas (no es el caso de los programas y juegos, para los que se aplica el concepto de copia de seguridad). No obstante, la difusión pública de contenidos protegidos con *copyright* sin autorización sí que constituye un ilícito civil y puede dar lugar a actuaciones judiciales que concluyan con la imposición de multas o indemnizaciones. Lo que está en el alero es la calificación de la puesta a disposición de obras en redes P2P como “difusión pública”: todavía, en el momento de redactar estos apuntes, no hay sentencia firme al respecto en nuestro país.

Últimamente, y debido a las fuertes presiones políticas de las multinacionales discográficas y cinematográficas, que exigen la promulgación de leyes que criminalicen a los usuarios del P2P, y también al hecho conocido de que muchas empresas se conectan a servidores P2P para monitorizar el tráfico de la red y recabar las direcciones IP de los ordenadores que están compartiendo determinados ficheros (eso sí, sin autorización judicial[31]), están empezando a utilizarse redes P2P totalmente descentralizadas y que utilizan técnicas criptográficas para proteger el anonimato de los usuarios (léase direcciones IP) y dificultar la identificación de los datos que circulan por la red.

El problema con estas redes es que, de momento, tienen pocos usuarios, lo cual hace que la red P2P no funcione bien globalmente. La fuerza del P2P radica precisamente en el número de usuarios: cuantos más usuarios haya en la red, habrá más fuentes a disposición de los usuarios y habrá un mayor número de ellos que descargarán de forma compartida los ficheros.

[31]Lo cual podría considerarse igualmente algún tipo de “piratería”.

Para concluir, y dejando de lado esta polémica, hemos de decir que el P2P es beneficioso para la sociedad y para el funcionamiento global de Internet: los procesos de descarga compartida y descentralizada contribuyen a una difusión más rápida de la información y a evitar colapsos y esperas en los servidores.

Tomemos como ejemplo el proceso (totalmente legal, por otra parte) de descarga de muchas distribuciones del sistema operativo Linux (Ubuntu, Debian, OpenSuSe, Mandriva...): estas distribuciones dejan imágenes ISO[32] de los CD o DVD de instalación a disposición pública en sus servidores. Cuando se lanzan las últimas versiones, hay millones de usuarios que se precipitan en su descarga, colapsando los servidores. Pues bien, para evitar los problemas de la descarga directa, todas estas distribuciones dejan ficheros en servidores *e-Donkey* o BitTorrent que pueden descargarse a través de clientes P2P.

Compartición de discos

Otro servicio interesante que proporciona Internet consiste en la posibilidad de poder compartir discos enteros o bien algunas de sus partes (particiones completas o directorios específicos) entre varios usuarios conectados a la red y autorizados a tal fin. Esto es algo frecuentemente necesario en las redes locales de las empresas o Universidades, donde se forman grupos de trabajo para desarrollar un proyecto determinado y se necesita acceso a ficheros y documentos comunes.

Para ello, se dispone de un ordenador que hará de *servidor de disco* ofreciendo (un espacio de) su disco duro para que pueda ser usado como un disco duro virtual a nuevos usuarios conectados a través de la red (y que tengan los permisos pertinentes). Conviene no confundir este servicio con el de *P2P file sharing* que hemos visto en la sección anterior, ya que aquí se comparten (partes de) discos y el servidor sí que contiene físicamente los ficheros que se desea compartir. Esta posibilidad la proporcionan una serie de servicios, entre los que destacamos:

- NFS (acrónimo inglés de *Network File System*) o sistema de ficheros en red en entornos UNIX (incluyendo a Linux, desde luego).
- SMB (acrónimo inglés de *Server Message Block*), también conocido como NetBIOS, en entornos Windows.

Ambos requieren especificar en la configuración del programa servidor[33] qué recursos se desea compartir, de qué manera y quién va a estar autorizado a compartirlos –normalmente indicando cuáles son las direcciones IP o nombres de las máquinas autorizadas para acceder al recurso compartido. En el caso de los programas cliente hay que configurar adecuadamente el acceso a estos recursos, especificando cuál es la dirección IP o nombre del servidor en cuestión.

Si queremos centralizar los recursos compartidos en un servidor con sistema operativo Linux y deseamos acceder a ellos mediante clientes que usan sistemas operativos

[32]Las imágenes ISO son copias binarias del contenido de un CD o DVD, que se pueden grabar directamente en un soporte virgen mediante cualquier programa de grabación conocido.

[33]En estos casos, normalmente, el programa forma parte del sistema operativo como un módulo más.

de la familia Windows –algo bastante frecuente, por otra parte– habremos de configurar el protocolo SMB sobre Linux.

La solución consiste en usar el programa *samba*, `http://www.samba.org`, disponible en todas las distribuciones más populares de Linux. *Samba* permite compartir recursos con máquinas Windows desde servidores UNIX, ya que implementa el protocolo SMB (el original junto con las extensiones de Microsoft) que éstas entienden.

Pero *samba* no sólo permite compartir recursos con máquinas Windows, sino también con máquinas UNIX. Por ejemplo, la red de aulas informáticas usa *samba* en el "Linux de aules" para realizar la identificación de los usuarios y exportar el disco (la partición) donde se encuentran las cuentas de usuario de `anubis`, el servidor de alumnos de la UJI (y donde se encuentra el disco que guarda todos vuestros ficheros).

Internet Relay Chat **(IRC)**

El servicio de IRC (acrónimo inglés de *Internet Relay Chat*), más comúnmente conocido como *chat*, permite crear un foro de debate interactivo al que acceden simultáneamente varios usuarios. En la práctica, la mayoría de servidores de IRC crean una serie de canales virtuales, o salas de charla (*rooms*), para separar las distintas temáticas sobre las que se puede dialogar.

Este servicio, por tanto, permite establecer un proceso de comunicación interactiva mediante texto (mensajes) entre las diferentes personas conectadas al servidor de IRC, a través de la utilización de los programas cliente adecuados (existen varios para Linux, Windows, Mac...).

A diferencia de otros mecanismos de comunicación interactiva, tales como la mensajería instantánea, la "charla" es en tiempo real: a medida que un usuario escribe datos en su ventana, los otros usuarios van leyendo y viceversa.

En la actualidad, muchos portales y pasarelas de Internet (de empresas proveedoras de Internet, periódicos en su edición electrónica, televisiones, etc.) ofrecen servicios de este tipo integrados en su web. En este caso, los usuarios están restringidos y limitados a lo que les permita la configuración establecida.

5.5.4. *Software* del cliente

Como ya sabemos, el modelo cliente-servidor en el que se basa Internet implica que se deban instalar programas que ofrezcan y gestionen servicios como los que hemos visto en las secciones anteriores en los ordenadores servidores conectados a la red.

Desde el otro punto de vista, tenemos que los ordenadores que deseen acceder a los servicios que ofrecen otras máquinas (los servidores) deben usar programas específicos que permitan hacerlo: son los programas cliente que se encargan de establecer el diálogo adecuado, siguiendo las órdenes del usuario, con los programas que gestionan el servicio en el servidor. Veamos algunos ejemplos ya para concluir con la sección 5.5:

- Los navegadores de páginas web son los clientes del servicio WWW. Hay un montón y para diversas arquitecturas: *Mozilla Firefox*, *Konqueror*, *Opera*, *Galeón*, *Internet Explorer*... Algunos, como *Firefox* y *Opera*, son multiplataforma

(hay versiones para Windows y Linux, por ejemplo). Normalmente los navegadores suelen dar soporte a otros servicios aparte de HTTP, tales como HTTPS, FTP o NEWS.

- Aparte de los navegadores web, existen una serie de programas específicos que son clientes del servicio de FTP (transferencia de ficheros). Este tipo de programas son más completos y ofrecen más funcionalidades, permitiendo una interacción total (dan soporte a todas las órdenes) con el servidor. En entornos UNIX (como Linux) podemos encontrar programas en "modo texto" (se ejecutan desde la consola) tales como *ftp*, *lftp* o *sftp* (los dos últimos ofrecen la posibilidad de establecer una conexión segura con un servidor que lo soporte empleando SSL). También existen varios programas gráficos con interfaces más intuitivas y menús contextuales que permiten realizar cualquier operación deseada de forma sencilla. Por ejemplo, tenemos *kasablanca* para el entorno de escritorio KDE de Linux, *gftp* para el entorno de escritorio GNOME de Linux, y *FileZilla* tanto para Windows como para Linux (es multiplataforma).

- A la hora de establecer una conexión remota segura (empleando técnicas criptográficas) o bien realizar transferencias seguras de datos con un servidor existen varias posibilidades. Tenemos clientes para consola ("modo texto") que son multiplataforma tales como *ssh* y *scp*, y también clientes gráficos (con menús y una interfaz más intuitiva) tales como *WINscp* (para sistemas Windows) o *Putty* (con versiones para Windows y Linux, por tanto multiplataforma).

- Si deseamos gestionar nuestro buzón de correo electrónico, archivar los mensajes, redactar nuevos, responder a los recibidos, efectuar búsquedas según un patrón determinado, etc., existen multitud de programas, tanto gráficos como de consola, que permiten realizar estas tareas y "dialogar" adecuadamente con el servidor de correo, con soporte, además, para protocolos complementarios tales como POP o IMAP. En el mundo Linux son muy populares programas como *mutt* o *pine*, que se ejecutan desde consola y son completísimos. Como clientes gráficos populares tenemos *Evolution* y *Thunderbird* (de la fundación Mozilla) que son multiplataforma y que incluyen componentes adicionales tales como agenda o lector de noticias (NEWS) incorporado. Además, podemos encontrar un programa gráfico más "ligero", *KMail*, para el entorno gráfico KDE de Linux. Para sistemas Windows, el programa más utilizado es *MS Outlook*. La mayoría de clientes de correo incluyen soporte integrado para filtros anti-SPAM (el término *spam*, en inglés un tipo de carne enlatada compuesta fundamentalmente de la del cerdo, sirve de denominación común para todos aquellos mensajes de correo no solicitado que llegan a nuestro buzón y se envían de forma masiva –propaganda, estafas. . .).

- Para el servicio P2P de compartición de ficheros, nos encontramos con que los dos protocolos más populares y con mayor número de usuarios son los de las redes *e-Donkey* y *BitTorrent*. Los programas más utilizados son, para la red *e-Donkey*, *eMule* (sólo para sistemas Windows) y *aMule* ("clon" del anterior y multiplataforma, ya que hay versiones para Windows y Linux); y para la red

BitTorrent el programa *Azureus*, que es multiplataforma.

- Existen multitud de clientes para los otros servicios, populares y menos populares, de Internet. Pongamos por caso, por ejemplo, los potentes gestores de bases de datos corporativas, que permiten acceso desde la red: así, tenemos clientes para *MySQL*, *PostgreSQL* y *Oracle*.

5.6. PROVEEDORES DE SERVICIOS DE INTERNET

En la actualidad, hay una serie de redes en nuestro país, tanto públicas como privadas, que están conectadas de algún modo –bien directa o indirectamente, a través de otras redes– a la red mundial de ordenadores y que, por tanto, proporcionan acceso a Internet a sus usuarios.

Un ejemplo de red pública es la red que agrupa a las redes de las distintas universidades españolas y que se conoce como REDIRIS. Esta red proporciona acceso a Internet a todos aquellos ordenadores conectados a las redes de cada universidad que hayan sido autorizados, o debidamente configurados, para ello por los respectivos administradores de cada red.

Pero también hay redes privadas, pertenecientes habitualmente a grandes empresas, corporaciones o multinacionales, como es el caso de la red perteneciente a la multinacional Telefónica. Algunas de estas redes se caracterizan por el hecho de permitir el acceso a usuarios particulares a cambio de un precio o tarifa determinado. Estas empresas dedicadas a comercializar el uso de su propia red privada para el acceso a Internet con unas condiciones de uso determinadas se denominan *proveedores de Internet*. Existen empresas de estas características tanto en nuestro país como en muchos otros.

En España operan varias de ellas, cada una con su propia infraestructura: Telefónica, Ono, JazzTel, etc. Mayoritariamente se ofrecen dos tipos de soporte para la conexión a la red: ADSL (en cualquiera de sus modalidades, véase la sección 5.4.1) y cable (véase igualmente la sección 5.4.1). De forma grosera, podemos decir, pues, que hay dos maneras de acceder a Internet:

- Mediante una *conexión permanente*, como la que se usa en máquinas de empresas e instituciones que tienen su propio dominio en Internet y, por tanto, su propio rango de direcciones IP a administrar. Dichas máquinas tienen un número de IP asignado que suele ser fijo. En la práctica, suelen estar permanentemente conectadas a Internet (salvo cuando están apagadas). Es el caso, por ejemplo, de las redes públicas de las universidades españolas.

- Mediante una *conexión temporal*, como la que se usa en determinados departamentos de empresas e instituciones (laboratorios, oficinas, etc.) y en conexiones domésticas. Son máquinas sin un número de IP fijo: cada vez que se conectan a la red obtienen un nuevo número –por ejemplo, a través de un servicio DHCP adecuadamente configurado por los administradores de la red en cuestión.

En cualquier caso, para poder efectuar una conexión a Internet desde el hogar se

debe contratar los servicios de una empresa de las características que hemos mencionado anteriormente, es decir, de un *proveedor de servicios de Internet* o ISP (acrónimo del inglés *Internet Service Provider*). De esta manera, cuando se desea efectuar la conexión a Internet desde casa, en realidad, se entabla una comunicación con el ISP. Una vez conectados a la red del proveedor, ya estamos conectados a Internet, pues su red ya lo estaba. Hay varios medios de establecer la comunicación con la red del ISP, entre los que destacan:

- Mediante módem telefónico (RTC), llamando a un cierto número de teléfono que nos indica el proveedor. Si bien hasta finales del siglo pasado y principios de éste, ésta era la forma más utilizada y extendida de conexión, en la actualidad está ya obsoleta y muy pocos proveedores la ofrecen.
- Con cualquiera de las variedades de ADSL existentes en el mercado, para lo que se requiere un módem ADSL adecuado (el que soporte la variante ofrecida). El ADSL es, en la actualidad, el medio de conexión más ampliamente utilizado en los países occidentales (el nuestro incluido).
- A través de una red de cable, empleando un cable-módem que sea compatible con el protocolo DOCSIS que usa el proveedor (que será una empresa con tendido de cable). Tras el ADSL, el cable es la modalidad de conexión casera a Internet que más se emplea.

Si la conexión desde el hogar es aceptada por el proveedor –por ejemplo, porque estamos al corriente de pago– nuestro ordenador recibe una dirección IP hasta el final de la sesión (cuando desconectamos). Esta dirección IP puede tener que ser renovada a intervalos fijos de tiempo (si estamos conectados muchas horas o incluso días enteros sin interrupción), normalmente por la misma, aunque la dirección IP puede cambiar durante alguna de las "renovaciones".

En algunas ocasiones, el proveedor de Internet puede exigir que, al establecer la conexión, el usuario envíe algún tipo de identificación, normalmente mediante el empleo de un nombre de usuario o *login* y una contraseña. Normalmente esto no suele hacerse hoy en día, ya que los proveedores tienen otras formas de identificar una conexión (mediante el número del módem, o el identificador de la línea, o la terminal, etc.). En principio, el ISP sólo proporciona la conexión a Internet, pero es frecuente que se ofrezcan servicios de valor añadido de forma gratuita, con el fin de atraer a un mayor número de potenciales usuarios:

- Varios buzones de correo electrónico (obviamente cada uno de ellos lleva aparejada la correspondiente dirección), aunque, eso sí, con un límite de espacio en disco determinado. Para poder gestionar estos buzones, el ISP suele ofrecer algún tipo de pasarela web (Webmail) o bien algún servicio gratuito de POP o IMAP (o incluso las tres cosas simultáneamente para que el usuario decida qué es lo que más le conviene).
- La posibilidad de publicar una página web personal. Habitualmente, las únicas restricciones que ponen los proveedores tienen que ver con que no se sobrepase una cantidad determinada de espacio en disco (entre 10 y 25 MB, normalmente)

y con que los contenidos se ajusten a ciertas normas de "etiqueta" (no valen temas "polémicos", contenidos que violen el *copyright* de alguien, insultos, etc.).

- Portales de noticias, música y vídeos gratis, y buscadores o directorios para facilitar la navegación.

Si se utiliza un módem telefónico convencional (RTC) para conectarse a Internet, hay que tener en cuenta que debe usarse un protocolo especial: PPP (acrónimo inglés de *Point-to-Point Protocol*, protocolo punto a punto en castellano). En este caso, no puede usarse el protocolo IP directamente y, para poder emplearlo (algo que resulta imprescindible en Internet, como ya sabemos), es necesario utilizar PPP en la conexión con el módem receptor del ISP (el que atiende la llamada telefónica de nuestro módem).

También es interesante saber que hay una variante de este protocolo que también se utiliza en algunas conexiones vía ADSL, dependiendo del proveedor y del dispositivo que empleemos (tipo de módem ADSL o bien si se trata de una combinación de tarjeta *ethernet* más *router* ADSL). En este caso, se trata de una variante denominada PPPoE (acrónimo inglés de *Point-to-Point Protocol over Ethernet*).

El protocolo PPP permite exigir la identificación del cliente mediante una contraseña. Tiene merecida fama de problemático, pues cada ISP puede tener diferentes exigencias a la hora de identificar a los clientes y la configuración de la conexión no resulta trivial –mucha jerga, ficheros de configuración complejos, mensajes informativos crípticos.... Por ejemplo, se pueden usar, básicamente, dos mecanismos distintos de identificación, PAP (acrónimo inglés de *Password Authentication Protocol*) y CHAP (acrónimo inglés de *Challenge Handshake Authentication Protocol*), y hay varias formas de configurarlos.

De todas formas, como ya hemos dicho, la utilización del módem RTC para efectuar la conexión a Internet está ya francamente en desuso. En cualquier caso, bien se emplee la línea telefónica convencional o bien se emplee cualquier variedad de ADSL o cable, lo más habitual hoy en día, se necesita facilitar una serie de datos al configurar una conexión con el ISP –algunos de los cuales ya podemos intuir, tras la atenta lectura de las secciones anteriores:

- ¿Se tiene una dirección IP fija o se va a obtener ésta dinámicamente? Recordemos que el servicio DHCP, entre otras cosas, permite asignar una dirección IP de forma automática y bajo demanda (es decir, dinámicamente).
- ¿Cuál es la dirección IP de la máquina que hace de pasarela (*gateway*) a la red del ISP? Para configurar la conexión a la red el sistema operativo tiene que conocer este dato, necesario para el correcto funcionamiento de los protocolos TCP/IP.
- ¿Cuál es la "máscara de red"? Éste es un dato requerido para el buen funcionamiento del protocolo IP.
- ¿Qué direcciones IP tienen las máquinas que ofrecen el servicio de nombres (DNS) en el ISP?

El proveedor de Internet (ISP) debe proporcionar esta información al usuario de alguna manera. Antiguamente, estos datos se proporcionaban a la firma del contrato de

prestación de servicios. Después, los proveedores pasaron a dar un CD-ROM con un programa de auto-configuración, pero que habitualmente sólo funcionaba en Windows. Con ello, consiguieron que los usuarios de otros sistemas operativos alternativos (como Linux) tuviesen que padecer la molestia de tener que llamar al servicio técnico y obtener estos datos después de un buen rato de charla inútil con comerciales absolutamente legos en el tema de las redes informáticas. En la actualidad, y para evitar molestias tanto a usuarios como a ellos mismos, la mayoría de los proveedores usa el *servicio DHCP* para proporcionar automáticamente todos estos datos a nuestro ordenador en el momento de establecer la conexión –con lo que simultáneamente se configura ésta.

5.6.1. Proveedores de Internet en España

En los primeros años de acceso público a Internet en nuestro país, desde 1996 hasta 1999[34], existían muchísimos pequeños proveedores de acceso. Todos ellos ofrecían acceso mediante la línea telefónica convencional (RTC) y cobraban una cuota mensual, normalmente entre 6 € y 20 €, tan sólo por proporcionar el servicio de acceso. El usuario debía pagar, aparte, la duración de la llamada telefónica local desde su módem RTC al del proveedor, o sea, todo el tiempo que se estaba conectado a Internet.

Como puede apreciar el lector, esto constituía un abuso. El precio que se pagaba por la "llamada del módem" (el tiempo de conexión a Internet) era el mismo que el de una llamada de voz. Téngase en cuenta que se utilizan diferentes mecanismos según se trate de una llamada de voz o de envío/recepción de datos. En el caso de usar voz para establecer la llamada *se conmutan circuitos*, es decir, hay una línea dedicada a establecer la comunicación. Para el envío de datos se utiliza un mecanismo de *conmutación de "paquetes"*: nuestros paquetes de datos se juntan con los de otros usuarios en la misma línea telefónica, es decir no hay una serie de circuitos dedicados a establecer una comunicación entre nosotros y el proveedor, sino que es la propia red telefónica la que da soporte a todos los usuarios de forma simultánea.

Posteriormente, aparecieron muchos proveedores que ofrecían acceso gratuito (tan sólo había que pagar el coste de las llamadas y, a veces, a un precio reducido). ¿Por qué ocurrió esto? Se trataba de empresas más grandes que lo hicieron fundamentalmente por tres razones (alguna de ellas poco ética):

- Para tener un medio que sirviese para dar a conocer y publicitar otros servicios de estas grandes empresas.
- Para hacer *dumping*[35] y así hundir a los pequeñas empresas proveedoras de Internet.
- Porque la empresa grande cobra parte o toda la conexión telefónica y le compensa sobradamente.

[34] Ya hubo algunos servicios pioneros a principios de los noventa, pero la incorporación a Internet fue tardía en nuestro país y no cabe considerarla como tal hasta mediados de los noventa.

[35] El *dumping* consiste en ofertar un producto por debajo de su coste real. La gran empresa puede asumir las pérdidas, por ejemplo con otros productos, ya que suele estar más diversificada en el mercado, pero la pequeña empresa no puede competir y se ve obligada a cerrar o a abandonar el mercado.

Después, comenzaron a ofertarse servicios de conexión mediante RDSI (Red Digital de Servicios Integrados). Esto fue una pequeña "estafa": RDSI no puede competir con ADSL y, curiosamente, la compañía que ofertaba RDSI de forma agresiva tenía ya comprada la tecnología ADSL. Su objetivo fue hacer rentable (para ellos) el RDSI durante un corto periodo de tiempo (un par de años) y después apostar por ADSL en detrimento de RDSI. El resultado fue que había bastantes clientes "enganchados" con RDSI que tuvieron que renovar sus inversiones si querían pasar a ADSL (los adaptadores de RDSI para ADSL no permitían realmente sacarle todo el partido a la conexión).

Paralelamente, surgieron las primeras "tarifas planas" de conexión a Internet ante las quejas de los usuarios por el hecho de que se cobrase igual una llamada de voz que una "llamada de módem" (ya hemos comentado al respecto en un párrafo anterior). La tarifa plana consistía en pagar una única cuota mensual fija para un número ilimitado de llamadas "de módem a módem". Esto lo podía controlar fácilmente el proveedor ya que los números a los que llamaban los usuarios correspondían a sus módems.

Desgraciadamente no todas las tarifas que se publicitaban como planas lo eran (¡ya surgió la típica picaresca española!). Algunas de ellas se ofrecían sólo para horarios restringidos: por ejemplo, a partir de las 18 h. en los días laborables. De ahí que muchos usuarios las bautizaran, socarronamente, como "tarifas onduladas". Aun hoy en día se ofrecen este tipo de tarifas planas falsas en algunas modalidades baratas de ADSL.

Otra posibilidad para conectarse a Internet consistía en contratar *bonos* con un número limitado de horas de utilización (10 horas, 20 horas. . .) con o sin restricciones horarias (menos o más caros, respectivamente).

Finalmente, surgieron a principios de siglo las conexiones a Internet *de alta velocidad* mediante ADSL y las redes de cable. Todo ello ha contribuido a que, en la actualidad, los proveedores de Internet en España sean mayoritariamente compañías de telecomunicaciones que participan también en negocios de transmisión de voz (telefonía fija y móvil) y comunicaciones (televisión a la carta, por ejemplo).

Estas compañías ofrecen fundamentalmente acceso a Internet a través de ADSL y conexión a redes de cable. Normalmente, estos servicios se ofrecen en zonas urbanas y metropolitanas, y también en grandes pueblos (con un núcleo de población importante). Existen modalidades de ADSL rural (a baja velocidad) para pueblos más pequeños. En la actualidad, ya hay compañías que ofrecen acceso por radiofrecuencia (Wi-fi) en algunas metrópolis.

Tal como hemos comentado anteriormente el acceso a través de la línea telefónica convencional (RTC) está casi "muerto" en la actualidad. No obstante, existe una solución para el acceso "ocasional" a Internet a través de un portátil y desde cualquier punto: las tarjetas 3G (véase la sección 5.4.1).

Como recordará el lector, estas tarjetas, son, en realidad, un teléfono móvil que se conecta a la red UMTS (3G) y proporcionan hasta 2 Mbps. aproximadamente. Se conectan al portátil mediante un puerto PCMCIA (específico de portátiles). Hay varias compañías de telefonía móvil (Orange, Movistar, Vodafone. . .) que ofrecen este servicio donde, habitualmente, se paga por volumen de datos transmitidos (en el momento de redactar estos apuntes 1 GB viene a suponer un desembolso de unos 30 € más o menos).

En España hay una serie de proveedores que operan a nivel nacional, como por ejemplo: Ono (ofrece conexión a su red de cable), Telefónica (ofrece conexión ADSL

y, de momento, todavía mantiene el acceso RTC), Terra (conexión mediante ADSL), Wanadoo (ofrece ADSL), Ya.com (ofrece ADSL) o Jazztel (oferta conexión mediante ADSL). También hay empresas que proporcionan servicios de cable o ADSL a nivel local o comarcal.

Se recomienda al lector interesado que consulte las posibles ofertas y sus precios en los teléfonos de atención al cliente y en las páginas web de los correspondientes operadores, ya que los precios sufren numerosas variaciones debidas a descuentos especiales y ofertas de temporada. Muchos operadores (Ono, Telefónica o Jazztel, por ejemplo) ofrecen, también, servicio telefónico y de televisión a la carta: existen "paquetes" con diversas combinaciones que pueden permitir un ahorro considerable si interesa contratar más de un servicio. Igualmente, se recomienda estudiar atentamente todas las condiciones del contrato, ya que:

- Hay que tener presente que en la "letra pequeña" de los contratos algunos proveedores fijan límites de descarga mensual, otros cobran por la instalación, otros cobran las llamadas efectuadas a partir de un minutaje (con RTC), etc.
- Algunas compañías ofrecen fraudulentamente una tarifa plana, cuando, en realidad, están ofreciendo una "tarifa ondulada": el minutaje se cobra o no en función del horario.

Por último, si el lector es usuario de otros sistemas operativos alternativos a los de la familia Windows –Linux, por ejemplo– se recomienda estudiar la compatibilidad de los módems y *routers* ofrecidos con el sistema usado. En muchas ocasiones ocurre que los dispositivos ofrecidos no funcionan con nuestro sistema operativo, pero hay otros que podemos adquirir por nuestra cuenta que sí que lo hacen y que, además, permiten la conexión con el soporte ofrecido por el proveedor sin mayor problema.

Es más, en muchas ocasiones los precios que ponen los proveedores a estos dispositivos son excesivamente onerosos, si bien es cierto que, en otras ocasiones, los ofrecen de forma gratuita (por estar en campaña o promoción). En cuanto a cuestiones de compatibilidad con los sistemas operativos se recomienda prestar atención a los *kits* automáticos de instalación, ya que casi siempre sólo están disponibles para Windows (y algunas veces son una clara señal de que el dispositivo sólo está soportado en dicho sistema operativo).

CAPÍTULO 6

EL INTÉRPRETE DE ÓRDENES UNIX

6.1. INTRODUCCIÓN

En el capítulo 2[1] aprendimos que podíamos realizar una serie de operaciones con ficheros y directorios: crear, copiar, borrar, cambiar su *ruta* en el sistema de ficheros[2], comprimir, modificar su contenido, listar su contenido, empaquetar, etc.

Al presentar el concepto de entorno de usuario comentamos que en los Sistemas Operativos (SS. OO.) de la "familia" Unix (como Linux) existen entornos orientados al trabajo en consola de texto, usando poderosos intérpretes de órdenes, los cuales son programas que leen las órdenes tecleadas por el usuario y controlan la ejecución de las mismas. Un buen administrador de sistemas debe saber desenvolverse con comodidad en este tipo de entornos: en muchas ocasiones deberá recurrir a la programación del intérprete de órdenes para automatizar muchas de sus labores (alta de usuarios, gestión de la red, análisis de los registros del sistema, copias de seguridad parciales y/o incrementales...).

Buscando en el baúl de los recuerdos...

Tal vez recuerdes, o hayas experimentado alguna vez con la ventana MS-DOS (Windows 95, Windows 98). Si es así, podrás hacerte una idea de cómo funcionan *grosso modo* los intérpretes Unix y su juego de órdenes básicas.

MS-DOS fue el S.O. precursor de los sistemas Windows y estaba totalmente orientado al trabajo en consola de texto.

continúa en la página siguiente ...

[1] "Sistemas Operativos y Entornos de Usuario".

[2] Recuerda que cambiar la ruta de un fichero o directorio puede implicar no sólo cambiar su ubicación en el sistema de ficheros, sino también cambiar su nombre y/o extensión.

...viene de la página anterior

De hecho, MS-DOS no fue más que una pobre y mala imitación de los poderosos sistemas Unix ya existentes: su juego de órdenes intentaba imitar las de los intérpretes Unix, pero carecía de varias de sus características y presentaba muchas limitaciones.

Obviamente, ello implica conocer (para empezar) el rico juego de órdenes que se proporcionan para realizar las operaciones básicas con ficheros y directorios. Observa que gran parte del potencial del intérprete de órdenes radica en su existencia.

En el capítulo 2 vimos que las interfaces gráficas de usuario permitían realizar de forma sencilla e intuitiva la mayoría de estas operaciones básicas. Para borrar un fichero, por ejemplo, basta con arrastrar el icono que lo representa y echarlo sobre el que representa una papelera en el escritorio. Sin embargo, borrar un fichero en un intérprete de órdenes Unix consiste en ejecutar la orden `rm` (*remove*): `rm nombrefichero`.

Llegado este punto, uno puede preguntarse "¿Por qué aprender a usar un intérprete de órdenes si existen sistemas con interfaz gráfico?". Para poder responder a esta pregunta, debes de tener en cuenta lo siguiente:

- Algunos sistemas profesionales no tienen un interfaz gráfico o éste no es estándar cuando, por contra, sí presentan juegos de órdenes estándar (o muy similares). Cuando los uses, necesitarás saber manejarte con el intérprete de órdenes.
- Los intérpretes de órdenes permiten automatizar ciertas tareas que son tediosas y propensas a la comisión de errores con interfaces gráficos. Ejemplos: elimina todos los ficheros del ordenador de tal tipo, haz una copia de seguridad de los ficheros modificados a partir de tal fecha, elimina los ficheros de tal directorio que no se han modificado en 15 días, etc.
- Los intérpretes de órdenes suelen incluir un potente lenguaje de programación (normalmente similar a C en cuanto a sintaxis y capacidad expresiva se refiere) que permite poder programar de forma automática tareas extraordinariamente complejas. Los programas realizados de esta manera se denominan *scripts*. Por ejemplo, muchos sistemas de copia de seguridad (*backup*) consisten en ejecutar una serie de *scripts* desde un intérprete de órdenes.

En este tema abordaremos el uso de las órdenes básicas de un entorno de línea de órdenes: el entorno de trabajo *Unix (Linux)*. Iremos repasando algunos conceptos sobre el sistema de ficheros vistos en el capítulo 2 conforme presentemos el entorno de trabajo Unix.

6.2. ÓRDENES BÁSICAS DEL INTÉRPRETE

Tal como hicimos en el capítulo 2, vamos a ilustrar la secuencia de operaciones básicas con ficheros y directorios mediante ejemplos de utilización, usando, en este caso las órdenes del intérprete Unix.

Refresquemos la memoria acerca del concepto de *ruta* dentro de la organización jerárquica de un sistema de ficheros[3]. Ya sabes que en Unix la jerarquía del sistema de ficheros tiene un nodo principal: el directorio raíz. El directorio raíz no tiene nombre y nos referimos a él con una barra (/).

Dentro del sistema de ficheros, cada fichero debe ser referenciado (nombrado) de forma *no ambigua*. Esto quiere decir que no podrá haber dos ficheros con el *mismo nombre absoluto* en el sistema de ficheros. Recuerda:

- *nombre absoluto* (ruta) de un fichero, es su nombre simple (nombre + extensión) precedido de la secuencia de directorios que hay que abrir desde la raíz para encontrarlo.

- *nombre relativo* a un directorio cualquiera X de un fichero, es su nombre simple precedido de la secuencia de directorios que llevan al fichero *desde* X.

Como explicamos en el capítulo 2, dentro del sistema de ficheros, el usuario siempre está trabajando en un directorio, al cual se le llama carpeta o directorio actual (también denominado directorio activo). El nombre relativo de un fichero siempre se interpreta como relativo al directorio actual.

Recuerda que si una ruta no empieza por / decimos que es una *ruta relativa*. En caso contrario es una *ruta absoluta*. Una ruta relativa añade implícitamente la ruta del directorio activo para formar así la ruta completa de un fichero.

6.2.1. Cambio de directorio activo

La orden `cd` (*change directory*) permite situarse en el directorio especificado, es decir cambiar el directorio activo. El directorio se puede especificar mediante la ruta completa o relativa al directorio de trabajo actual. Si utilizamos `cd ..` volvemos al directorio que incluye al que estamos actualmente situados (directorio "padre"). Por ejemplo, podemos cambiar el directorio activo a `/home` con:

```
$ cd /home
```

o al de otro usuario con:

```
$ cd /home/ana
```

Desde nuestro directorio personal podemos cambiar a `/home` con:

```
$ cd ..
```

Desde nuestro directorio personal podemos cambiar a `/home/ana` con:

```
$ cd ../ana
```

Desde nuestro directorio personal podemos cambiar a la raíz / con:

```
$ cd ../..
```

Finalmente, si ejecutamos `cd` sin una ruta, nos devuelve al directorio personal.

[3]Recuerda que mediante el uso de carpetas (o directorios) la información se organiza en forma de *árbol* (capítulo 2).

6.2.2. ¿Cuál es ahora el directorio activo?

La orden `pwd` (*path working directory*) muestra por pantalla el directorio activo actual (la ruta completa, es decir, su nombre absoluto):

```
$ pwd
/home/juan
$ cd ..
$ pwd
/home
$ cd
$ pwd
/home/juan
```

6.2.3. Listar el contenido de un directorio

La orden `ls` (*list*) permite listar el contenido de un directorio. Si no hay presente ningún argumento (aparte de las opciones que se pueden usar, como veremos a continuación) se supone como argumento predeterminado el directorio activo actual (que, como se dijo en el capítulo 2, se representa en Unix mediante ".").

```
$ ls /
bin    dev    etc    home   lib    mnt   tmp    usr    var
$ cd /
$ ls
bin    dev    etc    home   lib    mnt   tmp    usr    var
```

La opción `-l` (abreviatura de *long*) permite mostrar la información completa ("listado detallado") de lo que contiene el directorio.

```
$ ls -l /
drwxr-xr-x    2 root     root         2144 ago  5 09:20 bin
drwxr-xr-x   27 root     root        75264 sep 23 07:45 dev
drwxr-xr-x   53 root     root         6760 sep 23 07:45 etc
drwxr-xr-x    4 root     root         4096 mar 21  2002 home
drwxr-xr-x    6 root     root         2232 ago  5 09:21 lib
drwxr-xr-x    2 root     root           48 mar 21  2002 mnt
drwxrwxrwt   33 root     root         3528 sep 23 13:19 tmp
drwxr-xr-x   13 root     root          376 abr 25 21:32 usr
drwxr-xr-x   18 root     root          456 jun 27 09:53 var
```

Las órdenes Unix pueden modificar su comportamiento mediante *opciones*. Las opciones empiezan por un guión (`-`). El listado detallado da mucha información sobre cada fichero:

- los permisos de seguridad (el texto que presenta este aspecto: `drwxr-xr-x`),
- número de enlaces (no lo estudiaremos por ahora),

- nombre del propietario (en el ejemplo, el usuario `root`),
- nombre del grupo propietario (cada usuario pertenece a uno o más grupos; en el ejemplo, el grupo se llama igual que el usuario),
- el tamaño expresado en bytes (en el caso de ficheros de texto, un byte es un carácter; en el caso de directorios tiene un significado especial que no detallaremos),
- la fecha y hora de la última modificación,
- el nombre del fichero o directorio.

Un par de opciones interesantes...

La opción `-R` procede de forma *recursiva* (esto es, subdirectorio por subdirectorio hasta que no quede ninguno por explorar) a listar el contenido de los directorios y de sus subdirectorios.

La opción `-a` permite mostrar los ficheros ocultos (aquéllos cuyo nombre comienza por ".").

6.2.4. Creación de ficheros

Además de crear ficheros como resultado de trabajar con diversas aplicaciones, en Unix puedes crear ficheros vacíos empleando la orden denominada `touch`.

```
$ touch mifichero
$ ls -l
-rw-r--r--    1 juan  migrupo       0 sep 24 13:58 mifichero
```

Observa que el fichero recién creado no ocupa espacio (0 bytes): está vacío.

6.2.5. Creación de directorios

La orden `mkdir` (*make directory*) crea un nuevo directorio con el nombre que indiquemos. Si el nombre es *relativo*, se crea a partir del directorio activo actual.

La primera orden crea un directorio llamado `apuntes` en el directorio activo.

```
$ mkdir apuntes
$ ls -l
drwxr-xr-x    2 juan  migrupo    4096 sep 25 08:46 apuntes
-rw-r--r--    1 juan  migrupo       0 sep 24 13:58 mifichero
$ cd apuntes
$ pwd
/home/juan/apuntes
```

Y esta otra crea un directorio llamado `juan` en `/tmp`:

```
$ mkdir /tmp/juan
```

6.3. USUARIO Y GRUPO PROPIETARIOS

Tal como explicamos en el capítulo 2, se puede cambiar el propietario y grupo de un fichero si se tiene permiso para ello.

- La orden `chown` (*change owner*) cambia el propietario de un fichero.

  ```
  $ chown ana fichero.txt
  ```

- La orden `chgrp` (*change group*) cambia el grupo al que pertenece un fichero.

  ```
  $ chgrp alumnos fichero.txt
  ```

6.3.1. Permisos

En el capítulo 2, hablamos de permisos de ficheros y directorios en Unix. Échale un vistazo de nuevo a este tema y recuerda qué acciones básicas permitían controlar los permisos. Como vimos, éstos determinan no sólo qué acciones se pueden hacer y cuáles no con ficheros y/o directorios, sino también quién puede y quién no puede realizarlas. En esta sección, vamos a aprender cómo interpretar y cambiar los permisos mediante la ejecución de órdenes del intérprete. Así, cuando solicitamos el listado largo (`ls -l`) de un directorio se muestra, a mano izquierda, una serie de caracteres con este aspecto.

```
- rwx  rwx  rwx
      ↑    ↑    ↑
 usuario grupo otros
```

Observa que el primer carácter es un guión (`-`) para ficheros y una `d` para directorios. Cada grupo de tres caracteres indica si un permiso está concedido (aparece la letra `r`, `w` o `x`) o no (aparece un guión en el lugar correspondiente) al usuario, grupo y otros, respectivamente. Veamos algunos ejemplos para que te hagas una idea:

- Permisos de lectura y escritura para usuario y grupo, pero no para otros, en un fichero:
  `-rw-rw----`
- Permisos de lectura y escritura para usuario y sólo lectura para el grupo y otros, en un fichero:
  `-rw-r--r--`
- Permiso de ejecución y lectura para todos:
  `-r-xr-xr-x`

- Todos pueden hacer todo:
  `-rwxrwxrwx`
- El propietario y los usuarios del grupo pueden hacerlo todo, los otros nada:
  `-rwxrwx---`
- El propietario puede hacerlo todo, los demás sólo pueden leer y ejecutar:
  `-rwxr-xr-x`
- Puede que aquí nos hayamos equivocado con los permisos, ¿no crees?:
  `-------rwx`

La orden `chmod` (*change modifiers*) permite modificar los permisos de un fichero o directorio. Se puede aprender cómo se usa viendo unos pocos ejemplos. Imagina que tenemos un fichero llamado `mio` cuya información de permisos es `-rw-r--r--`:

- Dar permiso de lectura, escritura y ejecución a los miembros del grupo (la letra `g` es abreviatura de *group*, grupo en castellano):

  ```
  $ chmod g=rwx mio
  ```

  Queda así `-rw-rwxr--`.

- Dar permiso de ejecución al usuario (la letra `u` es abreviatura de *user*, usuario en castellano):

  ```
  $ chmod u+x mio
  ```

  Queda así `-rwxrwxr--`.

- Eliminar el permiso de lectura a otros (la letra `o` es abreviatura de *others*, otros en castellano):

  ```
  $ chmod o-r mio
  ```

  Queda así `-rwxrwx---`.

- Eliminar permiso de escritura a todos (la letra `a` es abreviatura de *all*, todos en castellano):

  ```
  $ chmod a-w mio
  ```

  Queda así `-r-xr-x---`.

- Dar permiso de escritura al usuario y al grupo:

  ```
  $ chmod ug+w mio
  ```

Queda así `-rwxrwx---`.

- Podemos combinar acciones distintas para varios grupos separando por comas. De esta manera, podemos quitar permiso de ejecución para usuario, quitar permisos de escritura y ejecución para grupo y añadir permiso de lectura para otros:

  ```
  $ chmod u-x,g-wx,o+r mio
  ```

  Queda así `-rw-r--r--`.

Sin embargo, a la hora de fijar todos los permisos *de una vez*, tal vez esta sintaxis te resulte más cómoda (en los ejemplos posteriores, supón que partimos de la siguiente situación de permisos para el fichero `mio`: `-rwxrwx---`):

- Si queremos poner sólo permisos de lectura y ejecución para el usuario y el grupo, y permiso de lectura para otros:

  ```
  $ chmod u=rx,g=rx,o=r mio
  ```

  Queda así `-r-xr-xr--`.

- Ahora, queremos poner permisos de lectura y de escritura para el usuario, de lectura para el grupo y *ningún* permiso para otros:

  ```
  $ chmod u=rw,g=r,o= mio
  ```

  Queda así `-rw-r-----`.

- Finalmente, dejaremos los permisos que originalmente presentaba el fichero:

  ```
  $ chmod o=r mio
  ```

  Queda así `-rw-r--r--`.

6.4. MÁS SOBRE RUTAS

Como ya vimos en el capítulo 2, podemos referirnos al directorio personal propio mediante el carácter ˜. Por ejemplo:

```
$ cd ˜
$ pwd
/home/juan
$ cd ˜/apuntes
$ pwd
/home/juan/apuntes
```

En ocasiones necesitarás referirte al directorio activo. Recuerda que puedes hacerlo usando un punto.

```
$ cd ~
$ ls .
apuntes   mifichero
$ cd ./apuntes
$ pwd
/home/juan/apuntes
```

6.5. OTRAS OPERACIONES CON FICHEROS Y DIRECTORIOS

En el intérprete, existen órdenes que permiten realizar diversas operaciones con ficheros y directorios:

- Copiarlos.
- Cambiarlos de ubicación ("renombrarlos").
- Borrarlos.
- Consultar el contenido de los ficheros de texto.

6.5.1. Copia de ficheros

La orden `cp` (*copy*) se utiliza para copiar ficheros (y carpetas enteras con todo su contenido). El número de opciones que presenta es muy elevado, así como las diversas formas en las que puede ser utilizada. Nosotros vamos a centrarnos en las operaciones de uso más frecuente. Se necesitan dos datos: la ruta del fichero original y la ruta de la copia.

Para copiar un fichero a otro: `cp fichero1 fichero2`. Esta orden crea una copia del fichero referenciado por la ruta `fichero1` al fichero referenciado por la ruta `fichero2`. En ambos casos se pueden usar rutas absolutas o relativas al directorio activo. Si `fichero2` no existe, se crea al hacer la copia y si existe el fichero, el contenido anterior se destruye al hacerla.

```
$ cp mifichero otrofichero
$ ls
apuntes  mifichero   otrofichero
```

Si la ruta de la copia (destino) identifica a un directorio existente en el sistema de ficheros, se crea una copia con el nombre original del fichero, pero dentro de dicho directorio.

```
$ cp mifichero apuntes
$ ls apuntes
mifichero
```

También podemos copiar un fichero a un directorio y, al mismo tiempo, cambiarle el nombre. Es decir, que efectuamos una copia del fichero con otro nombre.

```
$ cp mifichero apuntes/otrofichero
$ ls apuntes
mifichero  otrofichero
```

En principio, la orden `cp` sólo funciona con ficheros, no con directorios. La opción `-r` permite efectuar una *copia recursiva* de un directorio con todo su contenido. Sólo se ha de especificar el nombre (absoluto o relativo) de ambos directorios. Primero, el directorio cuyo contenido completo (ficheros y subdirectorios y los que éstos contienen) se desea copiar (debe existir), `dir1`. Después, el directorio donde se desea efectuar la copia, `dir2`. Si no existe, se crea. Si existe, se añade todo `dir1` como un subdirectorio más de `dir2`, sin destruir su contenido anterior.

```
$ cp -r apuntes nuevo
$ ls
apuntes mifichero nuevo otrofichero
$ ls apuntes
mifichero  otrofichero
$ ls nuevo
mifichero  otrofichero
```

6.5.2. Cambio de ubicación de ficheros y directorios

La orden `mv` (*move*) permite cambiar la ruta (nombre incluido) de un fichero o carpeta, lo que en algunos casos puede suponer cambiarlo de posición en el sistema de ficheros, simplemente cambiarle el nombre o bien ambas cosas a la vez. Recuerda: la orden `mv` funciona tanto con ficheros como con directorios.

Así, `mv origen destino` cambia el fichero (directorio) identificado por la ruta `origen` a la ruta especificada en `destino`. Los nombres pueden ser absolutos o relativos.

Si la ruta `destino` identifica a un directorio existente, `mv` mueve el fichero identificado por la ruta `origen` a ese directorio conservando su mismo nombre dentro del directorio.

```
$ mv otrofichero apuntes
$ ls
apuntes   mifichero   nuevo
$ cd apuntes
$ ls
mifichero   otrofichero
```

¡Ojo! Si ya existe un fichero con la ruta del segundo argumento, dicho fichero es sustituido por el primero, perdiéndose la información del segundo (como ha sido el caso en el ejemplo anterior). En estos casos, podemos hacer que `mv` pida confirmación usando la opción `-i`. También podemos cambiar el nombre del fichero al mismo tiempo que lo trasladamos de lugar.

```
$ mv otrofichero ../conotronombre
$ cd ..
$ ls
apuntes   conotronombre  mifichero  nuevo
```

O podemos, simplemente, cambiar el nombre del fichero.

```
$ mv conotronombre definitivo
$ ls
apuntes   definitivo   mifichero  nuevo
```

6.5.3. Eliminación de un fichero

La orden `rm` (*remove*) permite eliminar ficheros. Al igual que `cp` presenta muchas opciones y variedades.

```
$ cd ~
$ rm definitivo
$ ls
apuntes   mifichero   nuevo
```

¡Cuidado! La información de los ficheros borrados con `rm` no puede recuperarse.

6.5.4. Eliminación de un directorio

La orden `rmdir` (*remove directory*) elimina un directorio *si está vacío*. Si la ruta o nombre especificado no identifica a un directorio existente y vacío, se produce un error.

```
$ ls
apuntes   mifichero   nuevo
$ rmdir apuntes
rmdir: ‘apuntes’: El directorio no está vacío
$ ls apuntes
mifichero
$ rm apuntes/mifichero
$ rmdir apuntes
$ ls
mifichero   nuevo
```

Es posible que, según como esté configurado el sistema, el programa pida confirmación para borrar el fichero que deseamos eliminar. La opción `-f` permite ignorar este comportamiento (`rm -f`). Por ejemplo: `rm -f apuntes/mifichero`.

¿Y si queremos borrar un directorio y todo su contenido? Realmente no hace falta que el directorio esté vacío para borrarlo, ya que la opción `-r` primero elimina su contenido y después el directorio en sí. Por ejemplo: `rm -r apuntes/` borraría el directorio `apuntes` (y todo su contenido) aunque éste no estuviese vacío. Por supuesto podemos utilizar esta opción combinándola con `-f` para evitar las preguntas pidiendo confirmación (`rm -r`, `rm -r -f` o bien `rm -rf`).

6.5.5. Consultar el contenido de un fichero

Tal como vimos en el capítulo 2, podemos visualizar el contenido de un fichero de texto abriéndolo con el editor correspondiente, pero es una operación tan frecuente que Unix proporciona órdenes más cómodas y rápidas. La orden `cat` (*catenate*) muestra el contenido del fichero por la pantalla (la de la consola). El contenido de un fichero es visualizable en pantalla si es texto (ASCII, alguna de sus extensiones ISO-8859-? o Unicode, según esté configurado el sistema), ya que así es como lo "interpreta" `cat` (consulta el capítulo 3[4]).

```
$ cat novela.txt
Era un noche oscura y tormentosa. El perfil de la universidad
se dibujaba contra el fulgor de los relámpagos. Las siniestras
...
a casa.
```

Si visualizas el contenido de ficheros largos, éstos no caben en una pantalla, así que pasan rápidamente y acabas por ver sólo el final. La orden `less` permite visualizar el contenido "pantalla a pantalla".

Esta orden es lo que se llama un *paginador*, esto es, permite visualizar el contenido de un fichero página a página ("pantalla a pantalla"). Al igual que `cat`, interpreta el contenido del fichero como texto.

Al ejecutarlo, "quepa" o no el contenido del fichero en la pantalla, el programa se detiene y sólo muestra la parte del fichero que sí "cabe"[5]. Al final, aparece una línea con *dos puntos* (:) o bien la palabra `(END)` si ya se está visualizando el final del fichero. Si pulsamos la tecla Enter se visualiza la siguiente línea del fichero en la parte inferior de la pantalla. Pulsando la barra espaciadora (espacio en blanco) se visualiza la siguiente página del fichero en pantalla. El programa termina su ejecución al pulsar la tecla q.

La orden `less` permite ir hacia adelante y hacia detrás. Podemos desplazarnos a izquierda, derecha, arriba y abajo usando las teclas de los cursores. May + p permite desplazarse al principio del fichero. May + g permite desplazarse al final del fichero.

6.6. AYUDA: MANUALES EN LÍNEA

Unix presenta centenares de órdenes y es fácil que olvidemos qué hace cada una, cómo se usa y qué opciones presenta. La orden `man` (*manual*) permite consultar páginas de manual que describen cada orden (¡incluso el propio man!).

```
$ man ls
NOMBRE
       ls, dir, vdir - listan los contenidos de directorios

SINOPSIS
       ls [opciones] [fichero...]
       dir [fichero...]
       vdir [fichero...]
```

[4] "Codificación de la información"

[5] Si cabe entero, lo mostrará completo.

```
       ...

DESCRIPCIÓN
       El programa ls lista primero sus argumentos no directorios  fichero,  y
       luego para cada argumento directorio todos los ficheros susceptibles de
       listarse contenidos en dicho directorio. ...
...

OPCIONES DE GNU
       -1, --format=single-column
              Lista un fichero  por línea. Esto es lo predeterminado cuando la
              salida estándar no es una terminal.

       -a, --all
              Lista todos los ficheros en directorios, incluyendo todos los
              ficheros cuyos nombres comiencen por `.'.
...
```

6.7. COMODINES

Hasta ahora hemos aprendido a manipular ficheros y directorios de uno en uno. En ocasiones queremos actuar del mismo modo sobre un conjunto de ficheros; por ejemplo, borrar todos los ficheros de texto de un directorio o mostrar el contenido de todos los ficheros cuyo nombre empieza por la letra "a". Los intérpretes de órdenes Unix permiten usar comodines al especificar el nombre de uno o más ficheros.

- Asterisco (`*`): el asterisco significa cero o más caracteres. Por ejemplo, borrar todos los ficheros del directorio activo que acaban con extensión `.txt`:

  ```
  $ rm *.txt
  ```

  Borrar todos los ficheros del directorio activo que empiezan por la letra `a`:

  ```
  $ rm a*
  ```

  Borrar todos los ficheros del directorio activo que empiezan por la letra `a` y acaban con `.txt`:

  ```
  $ rm a*.txt
  ```

  Borrar todos los ficheros del directorio `/tmp`:

  ```
  $ rm /tmp/*
  ```

  Borrar todos los ficheros del directorio activo:

  ```
  $ rm *
  ```

- Interrogante (`?`): representa siempre *un único carácter* (que puede ser cualquiera). Por ejemplo, si en el directorio activo tuviésemos los ficheros `f1.txt`, `f2.txt`, `fa.txt`, `fm.txt` y `f.txt` la siguiente orden borraría a todos *excepto al último*:

  ```
  $ rm f?.txt
  $ ls
  f.txt
  ```

  Sin embargo, con esta orden se borrarían todos:

  ```
  $ rm f*.txt
  $ ls
  ```

Los comodines permiten trabajar con muchos ficheros a la vez, seleccionándolos con criterios bastante sofisticados. Los entornos gráficos suelen fallar, por ejemplo, en aspectos como expresar criterios similares para seleccionar cómodamente ficheros. El empleo de comodines permite expresar patrones bastante sofisticados en ocasiones: es lo que se conoce como expresiones regulares. Recuerda que otra ventaja importante de los intérpretes de órdenes es que son programables: es posible crear un fichero con una secuencia de órdenes que se ejecuta cada vez que lo deseamos (*script*).

6.8. OTRAS ÓRDENES ÚTILES

- `lpr` (*line printer*) manda un trabajo a impresora (con la opción `-P` se puede indicar a cuál si hay más de una configurada en el sistema).
- `quota` consulta la cuota de disco (la cantidad de megas que puedes usar en el disco duro) y el espacio disponible (úsala con la opción `-v` en `lynx` para saber en todo momento cómo estás de espacio en disco).
- `gzip` comprime un fichero. Crea un fichero con el mismo nombre y una extensión adicional (`.gz`). Si el fichero es de texto, puede reducirlo significativamente (por ejemplo, pasando a ocupar un 30 % del tamaño original).
- `gunzip`: descomprime ficheros comprimidos con `gzip`.
- `file` informa acerca del tipo de contenido de un fichero.
- `wc` (*word count*) cuenta el número de caracteres, palabras y líneas de un fichero de texto.
- `tar` (*tape archiver*) crea un único archivo que empaqueta varios ficheros y directorios en su interior, manteniendo la estructura de directorios que tuvieran en el momento de la creación del archivo. Es útil para hacer copias de múltiples ficheros y directorios. Igualmente, permite hacer copias de seguridad con cierta comodidad. Es compatible con empaquetadores para Windows tales como *WinZip* o *7-zip*. Para su utilización, nos centraremos en:

- `tar -cvzf archivo.tgz ruta-directorio` para comprimir y archivar en el fichero que llamaremos `archivo.tgz` todos los ficheros contenidos en el directorio identificado por `ruta-directorio`, incluyendo los contenidos de los subdirectorios y respetando la estructura existente en el momento de ejecutar la orden.
- `tar -tvzf archivo.tgz` para visualizar en pantalla el contenido de un archivo creado con `tar`.
- `tar -xvzf archivo.tgz` recupera los ficheros y subdirectorios contenidos en el fichero denominado `archivo.tgz`, manteniendo la estructura almacenada, en el directorio en el que nos encontremos.

Si eliminamos la letra `z` de las opciones no se realiza (des)compresión alguna, simplemente se archiva o se extrae.

APÉNDICES (no para examen)

Más utilidades para ficheros de texto

- `head` muestra las primeras líneas de un fichero (la "cabeza").

```
$ head -2 novela.txt
Era un noche oscura y tormentosa. El perfil de la universidad
se dibujaba contra el fulgor de los relámpagos. Las siniestras
```

- `tail` muestra las últimas líneas de un fichero (la "cola").

```
$ tail -1 novela.txt
a casa.
```

- `sort` muestra el contenido de un fichero ordenado alfabéticamente línea a línea:

```
$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
...
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash

$ sort /etc/passwd
adabas:x:36:100:Adabas-D database admin:/usr/lib/adabas:/bin/bash
amanda:x:37:6:Amanda admin:/var/lib/amanda:/bin/bash
amarzal:MxGwFkl878xxR:1001:1999:Andrés Marzal:/home/amarzal:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
...
wwwrun:x:30:65534:WWW daemon apache:/var/lib/wwwrun:/bin/bash
zope:x:64:2:Zope daemon:/var/lib/zope:/bin/false
```

- `grep` muestra las líneas de un fichero que contienen una palabra o fragmento de texto.

```
$ grep oscura novela.txt
Era un noche oscura y tormentosa. El perfil de la universidad
su zarpa oscura y putrefacta agarró al estudiante por el cuello

$ grep daemon /etc/passwd
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
wwwrun:x:30:65534:WWW daemon apache:/var/lib/wwwrun:/bin/bash
irc:x:39:65534:IRC daemon:/usr/lib/ircd:/bin/bash
zope:x:64:2:Zope daemon:/var/lib/zope:/bin/false
```

Redirección

El símbolo > *redirige la salida que produce la ejecución de un determinado programa (u orden) en la pantalla del ordenador a un fichero*, es decir, lo que debería salir en pantalla pasa a formar parte de un fichero. Por ejemplo, usando redirección de salida, podríamos crear un fichero que contuviese un listado de ficheros:

```
$ ls > contenido
$ ls
apuntes contenido mifichero
$ cat contenido
apuntes mifichero
```

Hemos visto antes que `cat` muestra el contenido de un fichero. Si lo usas sin ningún nombre de fichero, hace una cosa distinta: lee el texto que escribas por teclado y lo muestra por pantalla, línea a línea.

```
$ cat
un ejemplo que tecleamos
un ejemplo que tecleamos
para ver cómo
para ver cómo
cat muestra el texto.
cat muestra el texto.
^D
```

Para finalizar la ejecución de `cat` en estas circunstancias, hemos de pulsar la tecla de control y la tecla `D` simultáneamente (`^D`): así avisamos de que no vamos a teclear más texto. De hecho, con la redirección podemos usar `cat` como un editor de textos muy primitivo:

```
$ cat > unfichero
un ejemplo que tecleamos
para ver cómo
cat muestra el texto.
^D
$ cat unfichero
un ejemplo que tecleamos
para ver cómo
cat muestra el texto.
```

La redirección funciona con cualquier orden que muestre información en pantalla:

```
$ sort unfichero > ordenado
$ cat ordenado
cat muestra el texto.
para ver cómo
un ejemplo que tecleamos
```

Hemos aprendido a redireccionar la salida (la información que se muestra por pantalla). Se puede redireccionar también la entrada (la información que tecleamos), sólo que entonces usamos el símbolo <. En el ejemplo que se muestra a continuación, los datos de entrada para `cat` son las líneas de texto almacenadas en el fichero `ordenado`:

```
$ cat < ordenado > otrofichero
$ cat otrofichero
cat muestra el texto.
para ver cómo
un ejemplo que tecleamos
```

Una opción muy potente[6] consiste en *enlazar* la *salida* de un programa con la *entrada de otro*, formando una cadena o *tubería* a través de la cual un programa envía datos y el otro los recibe. Este tipo de enlace se denomina precisamente *tubería* (*pipe* en inglés) y se expresa con la barra vertical (`|`).

Por ejemplo, la orden `who` indica qué usuarios están conectados a un ordenador en un momento determinado, pero proporciona una lista desordenada. La podemos obtener ordenada combinando las "habilidades" de `who` y `sort` con una tubería:

```
$ who
tenaj    pts/1    Sep 25 07:45 (stargate.si.uji.es)
smarti   pts/2    Sep 25 09:15 (chaini.act.uji.es)
pepe     pts/0    Sep 25 08:48 (pepe.si.uji.es)
root     pts/4    Sep 24 09:12 (dentaku.si.uji.es)
barreda  pts/5    Sep 24 08:18 (fermi.act.uji.es)
vilata   pts/7    Sep 25 09:32 (mezzanine.act.uji.es)
llopis   pts/6    Sep 24 17:22 (ced0113.act.uji.es)
climente pts/11   Sep 25 09:49 (nobel.phys.uni.torun.pl)
jvilar   pts/13   Sep  9 11:00 (dafne.act.uji.es)
navarro  pts/12   Sep 24 15:05 (kiko.si.uji.es)
$ who | sort
barreda  pts/5    Sep 24 08:18 (fermi.act.uji.es)
climente pts/11   Sep 25 09:49 (nobel.phys.uni.torun.pl)
jvilar   pts/13   Sep  9 11:00 (dafne.act.uji.es)
llopis   pts/6    Sep 24 17:22 (ced0113.act.uji.es)
navarro  pts/12   Sep 24 15:05 (kiko.si.uji.es)
pepe     pts/0    Sep 25 08:48 (pepe.si.uji.es)
root     pts/4    Sep 24 09:12 (dentaku.si.uji.es)
```

[6] Y que se utiliza, frecuentemente, en la realización de *scripts* o programas del intérprete de órdenes.

```
smarti   pts/2    Sep 25 09:15 (chaini.act.uji.es)
tenaj    pts/1    Sep 25 07:45 (stargate.si.uji.es)
vilata   pts/7    Sep 25 09:32 (mezzanine.act.uji.es)
```

Y si nos interesase conocer sólo los tres primeros en orden alfabético, podríamos combinar la tubería actual con la orden `head` mediante una nueva tubería:

```
$ who | sort | head -3
barreda  pts/5    Sep 24 08:18 (fermi.act.uji.es)
climente pts/11   Sep 25 09:49 (nobel.phys.uni.torun.pl)
jvilar   pts/13   Sep  9 11:00 (dafne.act.uji.es)
```

Si, además, queremos almacenar esa información en un fichero, usamos redirección:

```
$ who | sort | head -3 > primeros
$ cat primeros
barreda  pts/5    Sep 24 08:18 (fermi.act.uji.es)
climente pts/11   Sep 25 09:49 (nobel.phys.uni.torun.pl)
jvilar   pts/13   Sep  9 11:00 (dafne.act.uji.es)
```

Observa que puedes obtener una funcionalidad similar a las tuberías utilizando redirección y un fichero auxiliar:

```
$ who > auxiliar
$ sort < auxiliar
```

MS-DOS, un S.O. antiguo, predecesor de Windows, "copió" el mecanismo de redirección de Unix. Sin embargo, utilizaba la "estrategia" descrita en el ejemplo anterior para recrear (imitar) el funcionamiento de las tuberías de Unix. Las tuberías de Unix ofrecen dos importantes ventajas:

- no hace falta un fichero auxiliar, así que se ahorra espacio en disco (memoria);
- los procesos implicados se ejecutan *simultáneamente* (es decir, el segundo comienza a ejecutarse antes de acabar el primero) comunicando directamente la salida de uno con la entrada del otro. En MS-DOS, el segundo comenzaba a ejecutarse sólo cuando había terminado el primero.

En Unix, la mayoría de las órdenes están concebidas como herramientas que luego pueden concatenarse mediante tuberías para crear otras más potentes.

CAPÍTULO 7

SEGURIDAD Y PROTECCIÓN DE LA INFORMACIÓN

7.1. CRIPTOGRAFÍA

La utilización de técnicas de criptografía tiene varios objetivos:

- "Esconder" información:
  - Procurar que sólo una persona, o grupo de personas, tenga acceso a cierta información.
  - Transmitir información por un canal más o menos seguro, sin que un tercero pueda leerla.
- Certificar origen: asegurar que la información proviene de quien se supone que proviene (un remitente determinado).
- Garantizar integridad: asegurar que la información que se recibe es la que se envió (no ha sufrido alteraciones).

Estos objetivos no son excluyentes y pueden lograrse por separado. Para conseguirlo se utilizan distintas técnicas. Nosotros trataremos los sistemas de clave privada, clave pública y PGP.

7.1.1. Sistemas de clave privada

En estos sistemas se utiliza una clave que sirve tanto para cifrar como para descifrar, técnica que se conoce como *cifrado simétrico*. La Figura 7.1 ilustra este mecanismo. Sus características más destacadas son las siguientes:

- La longitud de la clave es fundamental: si es corta, el método no es seguro; si es larga, es incómoda.

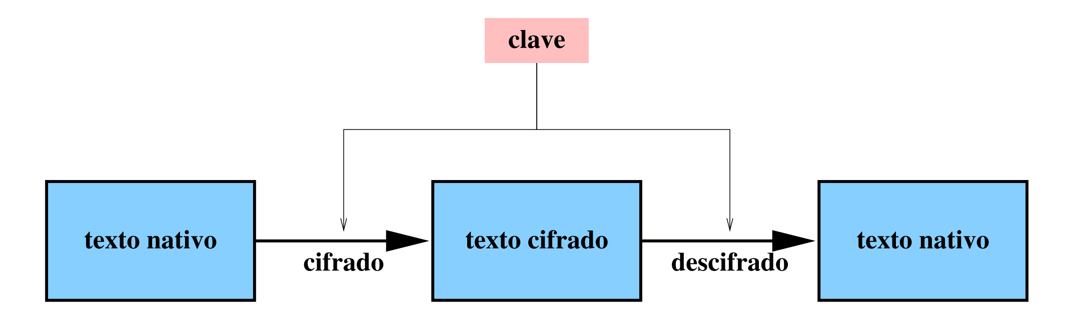


Figura 7.1. Mecanismo de funcionamiento de los sistemas de clave privada

- Es necesario que las dos partes conozcan la clave. Ello implica que hay que transmitirla de manera segura.

Un sistema popular de clave privada es el DES (Data Encription Standard). Problemas que se plantean:

- Como hemos dicho, las dos partes deben conocer la clave. ¿Cómo se transmite ésta de forma segura?
- Hay que mantener una clave distinta para cada usuario con el que deseamos intercambiar información confidencial.

7.1.2. Sistemas de clave pública

Para evitar el problema de tener que transmitir la clave privada de forma segura, se emplean los sistemas de clave pública o *cifrado asimétrico*. La ilustración de la Figura 7.2 muestra su mecanismo de funcionamiento.

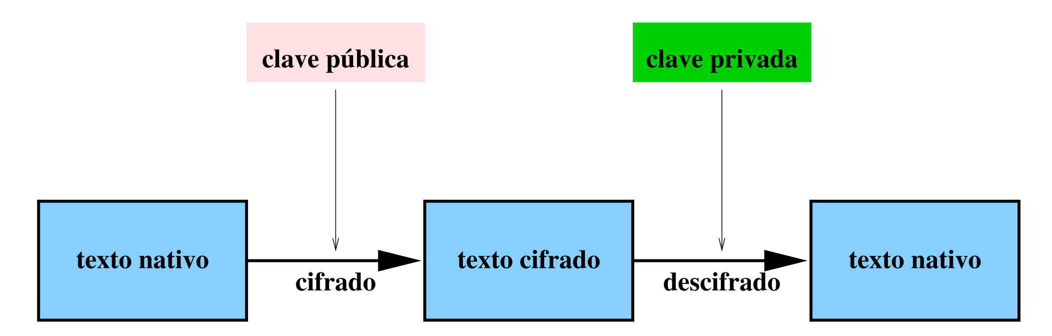


Figura 7.2. Mecanismo de funcionamiento de los sistemas de clave pública

Se emplean dos claves distintas por cada "persona", una para cifrar, que normalmente es la clave pública, y otra para descifrar, que normalmente suele ser la clave privada. La clave de cifrado es *pública*, esto es, conocida por todo el mundo; la de descifrado es *privada* y, por tanto, solamente conocida por su propietario. Ambas constituyen un *par de claves*. Veamos un ejemplo de funcionamiento: Juan y Carlos tienen un par de claves pública/privada;

- Juan obtiene la clave pública de Carlos, k.

- Juan escribe un mensaje M y lo cifra usando k.
- Juan envía el mensaje cifrado $C_k(M)$.
- Carlos recibe el mensaje y le aplica su clave privada k' obteniendo M.

En resumen, cualquiera puede cifrar un mensaje y enviárselo a Carlos, pero sólo él podrá descifrar (y por tanto leer) los mensajes que le llegan. Nadie más, ni siquiera Juan podrá descifrar el mensaje que acaba de cifrar. Ten en cuenta, no obstante, que ambas claves se pueden usar tanto para cifrar como para descifrar. De hecho, ambas claves *son complementarias*. Propiedades:

- Los mensajes cifrados con la clave pública de alguien sólo pueden descifrarse con su clave privada. En este caso, conseguimos la confidencialidad en el envío del mensaje (sólo el receptor puede descifrarlo con su clave privada).
- Los mensajes cifrados con la clave privada de alguien pueden descifrarse con su clave pública. En este caso perseguimos la certificación de la identidad del remitente. Cualquiera podrá descifrar el mensaje usando la clave pública de quien lo envió (conocida por todos), pero solamente ese alguien podría haber sido el autor de dicho mensaje.

Envío de mensajes cifrados

El proceso que hemos de seguir para enviar un mensaje cifrado a A cuando no queremos que nadie (salvo A) pueda acceder a su contenido es el siguiente:

- Ciframos el mensaje con la clave pública de A.
- A lo descifra con su clave privada.

Puesto que sólo A conoce su clave privada, sólo A puede descifrar el mensaje. Este método evita el problema de la transmisión de la clave: no hay problema en que la clave pública sea conocida. Además, permite comunicarse de manera segura con cualquiera que haya publicado su clave pública.

Uno de los métodos más conocidos de cifrado asimétrico es RSA (Rivest, Shamir y Adleman). Su gran inconveniente es que resulta 1.000 veces más lento que los sistemas de clave privada. Si queremos más rapidez podemos usar el sistema de clave privada para "tratar" el mensaje (generalmente mucho más largo) y el de clave pública para cifrar la clave privada (generalmente mucho más corta) que acabamos de usar.

Envío de mensaje cifrado y con certificación de identidad del remitente

En este caso, en la Figura 7.3 mostramos el esquema que habría que utilizar si A envía un mensaje a B que no desea que pueda ser leído por nadie (salvo B) y, al mismo tiempo, se desea que B pueda asegurarse de que, efectivamente, el mensaje proviene de A.

Nótese que A es la única persona que puede cifrar el mensaje en origen y enviarlo a B, ya que es el único que conoce (o debería conocer) su clave privada. Por otra parte, B es el único que puede descifrar en destino, en primera instancia, el mensaje recibido con su clave privada (que sólo él conoce).

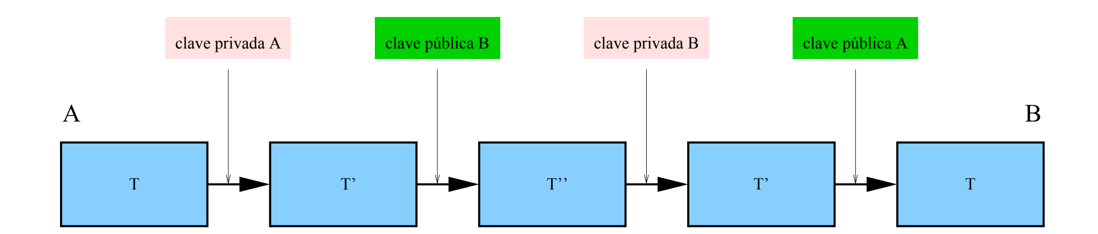


Figura 7.3. Procedimiento para enviar mensajes cifrados al mismo tiempo que garantizamos la identidad del emisor

Envío de mensaje cifrado pero legible por todos y con certificación de identidad del remitente

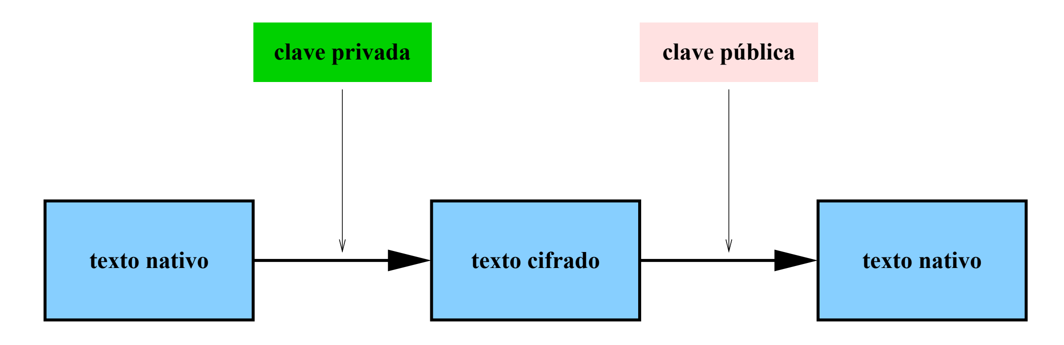


Figura 7.4. Procedimiento a seguir cuando sólo nos interesa realmente garantizar la identidad del emisor

Si solamente estamos interesados en la firma del mensaje (es decir, en garantizar que lo envía quien dice que lo envía), pero no nos importa que se pueda acceder a los contenidos del mensaje, entonces podemos adoptar el esquema, más sencillo, ilustrado en la Figura 7.4.

Realmente, la clave pública la conoce todo el mundo (por tanto, cualquiera puede descifrar y leer el mensaje), pero el único que pudo enviar el mensaje es el emisor, ya que es el único que conoce su clave privada.

Envío de mensaje "abierto" con garantía de integridad y certificación de remitente

La firma de un documento (no confundir con la firma personal) es un resumen cifrado del mismo (mucho más breve). El documento en sí no necesita ser cifrado (se puede añadir la firma al final del documento).

El resumen se cifra con la clave privada del remitente. El receptor sabe quién firmó (gracias a la clave pública) y esa persona no puede repudiar la autoría del mensaje. Características:

- No existe la posibilidad de volver a obtener el mensaje partiendo del resumen generado.
- Si se cambia el mensaje, el resumen que se obtiene es diferente.

Por tanto, si se cambia el contenido del mensaje, el que verifica la firma lo sabrá.

7.1.3. PGP

PGP (Pretty Good Privacy) es un programa que combina las ventajas de los sistemas de clave pública y los de clave privada. Veamos cómo funciona el mecanismo para el envío de mensajes cifrados con PGP:

- Se genera una clave aleatoria $\mathcal{C}$.
- Se utiliza la clave para cifrar el mensaje con un algoritmo de (sólo) clave privada.
- Se cifra $\mathcal{C}$ con una clave pública.
- Se envían conjuntamente la clave cifrada y el mensaje (igualmente cifrado).
- El receptor utiliza su clave privada para descifrar $\mathcal{C}$ y después usa $\mathcal{C}$ para recuperar el mensaje original.

PGP también ofrece la posibilidad de firmar digitalmente los mensajes para que el que lo recibe se asegure de que corresponde a lo escrito por el remitente:

- Se calcula un valor numérico a partir del mensaje que queremos enviar. Ningún otro mensaje tiene (en la práctica) dicho valor.
- Se cifra dicho número con la clave privada del emisor y se añade al final del mensaje.
- El texto no tiene porqué cifrarse si no se desea.
- El receptor descifra el número empleando la clave pública del emisor y comprueba que corresponde al del mensaje (tras calcularlo nuevamente al recibir dicho mensaje).

7.1.4. Certificados

Los métodos de clave pública tienen un problema al usarse para generar firmas: asegurarse de que la clave pública es de quien corresponde, es decir, de quien dice ser (garantía de identidad).

¿Sabías que. . . ?

Un certificado consta de tres partes:

- Una clave pública.
- Información acerca del propietario.
- Una firma electrónica.

La firma electrónica corresponde a una fuente que sea fiable: la *entidad certificadora*.

Si las comunicaciones se producen en un círculo reducido donde tenemos acceso físico a la persona, por ejemplo dentro de una empresa, no suele haber problemas;

podemos obtener la clave directamente de él (en un disquete, CD-ROM, de un servidor conocido, etc.), en cuyo caso la identidad, obviamente, está clara. En caso contrario, hemos de recurrir a una *entidad certificadora* (una fuente fiable) para obtener lo que se llama un *certificado*, el cual certifica la identidad de quien publica la clave.

7.1.5. Un ejemplo de uso de criptografía

La Agencia Tributaria, por ejemplo, usa un sistema criptográfico para que los contribuyentes puedan cumplimentar la declaración de renta por la red. El "certificado" de la Agencia Tributaria es un conjunto de datos que consta de:

- Identificación del titular del certificado.
- Distintivos del Certificado: número de serie, entidad que lo emitió, fecha de emisión, fecha de caducidad, etc.
- Una pareja de claves: pública y privada.
- La firma electrónica de la autoridad de certificación que lo emitió.

Los datos se dividen en:

- Parte privada: clave privada.
- Parte pública: resto de datos del certificado, incluida la firma electrónica de la autoridad de certificación que lo emitió.

El certificado permite:

- Identificar a las dos partes que se conectan de forma remota.
- Intercambiar una clave con la que se van a cifrar los datos que se transmiten entre los dos extremos y garantizar que esa clave no la conocen más que ellos.
- Firmar electrónicamente los datos que se envían, de tal forma que se pueda probar su integridad y procedencia.

La autoridad certificadora es la Fabrica Nacional de Moneda y Timbre (FNMT). La obtención del certificado por primera vez se realiza mediante dos conexiones con la FNMT, intercaladas con la acreditación personal del solicitante en una oficina de la AEAT para garantizar la identidad del que va a ser titular del certificado y evitar la suplantación de su personalidad.

La presentación personal y acreditación de identidad mediante DNI es, en este caso, la única forma que tiene la autoridad certificadora (FNMT) de poder garantizar que el certificado va a parar a las manos "correctas" (recuerda, se presenta el DNI).

7.2. SEGURIDAD Y PROTECCIÓN DEL PC

Hasta hace unos pocos años, los ordenadores personales eran, en general, equipos aislados a los que tan sólo se podía acceder sentándose frente a ellos, con lo cual los riesgos de accesos indeseados a un equipo eran fácilmente evitables. Sin embargo, recientemente, un mayor número de ordenadores personales tanto de uso privado como de uso en empresas o otros entornos laborales, están conectados a diferentes redes de ordenadores y en particular a *Internet*. Esto hace que el acceso a estos equipos y a los datos que contienen sea mucho más sencillo y rápido, al tiempo que los riesgos de accesos y usos indeseados se hayan incrementado enormemente. Internet es una fuente de peligros que pueden dañar seriamente los ordenadores de usuarios desprevenidos.

En esta sección vamos a plantear los principales riesgos que un usuario doméstico afronta y las formas en las que se les puede hacer frente. Haremos un especial énfasis en la problemática con usuarios de sistemas operativos de la familia Microsoft Windows por ser los más habituales, aunque no perderemos de vista los sistemas basados en Linux, que también tienen sus problemas. Tendrán, pues, estas explicaciones un enfoque eminentemente práctico.

Actualizaciones

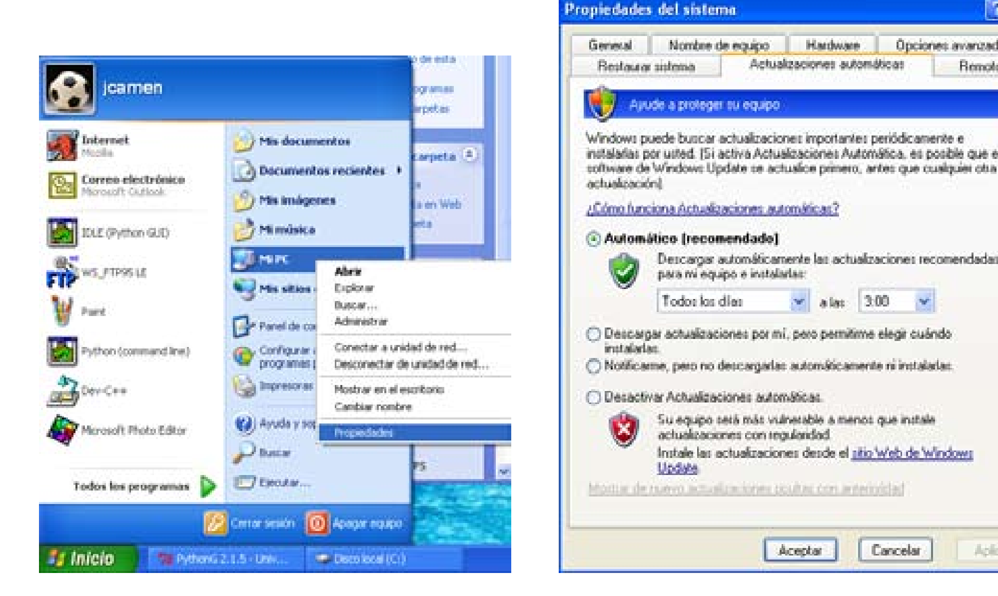
Antes de entrar a describir los distintos tipos de riesgos posibles, la primera medida de seguridad que se debe aplicar es la *actualización continua*, y si es posible de forma automática, de los componentes del Sistema Operativo (S.O.).

En este sentido, la aplicación *Windows Update* está disponible para Windows XP y la "familia" Windows 2000 (cuya última versión es Windows 2003 Server). Windows 98 y ME ya no tienen actualizaciones. Esta aplicación se conecta al sitio web de Microsoft y permite comprobar la existencia de parches de actualización del S.O., así como descargarlos e instalarlos en el equipo. Obviamente, se requiere una licencia válida del producto para poder disfrutar de estos beneficios.

Es posible programar esta utilidad para que la comprobación e instalación de actualizaciones se realice bien de forma automática al arrancar el ordenador, o bien de forma periódica. Todo ello se puede configurar accediendo a la pestaña *Actualizaciones automáticas* que aparece en la ventana de *Propiedades del sistema*. Esta ventana aparece al seleccionar la opción *Propiedades* del menú contextual asociado al icono *Mi PC* en el *menú Inicio*. La Figura 7.5 proporciona más detalles al respecto. De cualquier modo, siempre es posible comprobar la existencia de actualizaciones visitando el sitio web de Microsoft. Ésta es la única opción de actualización para aquellas versiones del sistema que no cuenten con la utilidad *Windows Update*.

También es posible encontrar actualizaciones para ciertas aplicaciones críticas como son *Internet Explorer* y la "familia" *Microsoft Office*, incluyendo *Outlook y Outlook Express*.

En el caso de los usuarios de Linux, es igualmente importante mantener actualizadas las diferentes aplicaciones, en particular aquéllas que ofrecen servicios de Internet, tales como servidores de correo, servidores de web, OpenSSH, etc. Existen herramientas para actualizar los diversos componentes y aplicaciones instalados en Linux: éstas pueden variar entre distribuciones (por ejemplo, YaST en SuSE), aunque también po-



(a) Menú contextual de *Mi PC* en el botón *Inicio*. (b) Pestaña para la configuración de *Actualizaciones automáticas*.

Figura 7.5. Configuración de *Windows Update*

demos encontrar herramientas gráficas que realizan esta función según el entorno gráfico de usuario con el que estemos trabajando. Por ejemplo, en el entorno KDE tenemos *adept* (y que es el que usa la distribución Kubuntu).

En las distribuciones Linux, es muy importante incluir los *repositorios de seguridad* para las actualizaciones críticas de los programas. Normalmente, los repositorios son servidores (WWW y/o FTP) "espejo"[1] que incluyen paquetes con programas listos para su instalación y configuración automática. Para poder usarlos, simplemente hemos de añadir sus direcciones (URL) en el apartado correspondiente de la configuración de la herramienta que usemos para gestionar las actualizaciones con nuestra distribución de Linux. Normalmente, esta información se almacena en un determinado fichero del sistema. Por ejemplo, en las distribuciones Debian y derivadas, tales como las de la "familia" Ubuntu (Ubuntu, Kubuntu, Xubuntu...), los datos de los repositorios se almacenan en el fichero denominado `sources.list` dentro del directorio `/etc/apt/`:

```
deb http://es.archive.ubuntu.com/ubuntu/ feisty main restricted
deb http://es.archive.ubuntu.com/ubuntu/ feisty-updates main restricted
deb http://es.archive.ubuntu.com/ubuntu/ feisty universe
deb http://es.archive.ubuntu.com/ubuntu/ feisty-updates universe
deb http://es.archive.ubuntu.com/ubuntu/ feisty multiverse
deb http://es.archive.ubuntu.com/ubuntu/ feisty-updates multiverse
deb http://security.ubuntu.com/ubuntu feisty-security main restricted
deb http://security.ubuntu.com/ubuntu feisty-security universe
deb http://security.ubuntu.com/ubuntu feisty-security multiverse
```

En este ejemplo, podemos ver la lista de repositorios para la versión *Feisty* de la distribución *Kubuntu*. En todos los casos, se trata de servidores WWW (protoco-

[1]Los servidores "espejo" son réplicas de uno (o varios) servidor(es) central(es).

lo `http`). Para poder actualizar y/o instalar *software* se requiere conexión a Internet, obviamente. En el texto, se han subrayado los repositorios de seguridad. Por si tienes curiosidad, es importante resaltar que los repositorios *universe* y *multiverse* son necesarios si se quiere instalar programas comerciales, no libres y/o herramientas multimedia complementarias que, por problemas legales, no pueden incluirse en el repositorio principal (*main*) de la distribución *Kubuntu.*

7.2.1. Virus

Una de las amenazas más serias es la contaminación por parte de virus informáticos. Desde que los ordenadores se convirtieron en una herramienta de trabajo habitual, los virus han estado siempre presentes amenazando el buen funcionamiento de los equipos informáticos. Con el desarrollo y la expansión de Internet, y las altas velocidades de transmisión y difusión que ésta ofrece, se ha convertido en el medio predilecto para la existencia de los virus. En la medida en que nuestros ordenadores estén más y mejor conectados a esta red de redes, el riesgo de ser afectado por algún virus se ha disparado.

Se suelen considerar como virus, programas que cumplen la mayor parte de estas características:

- No se presentan solos sino que se suelen adherir a un huésped. Estos huéspedes pueden ser otros programas, correos electrónicos, páginas web, macros, ficheros o volúmenes (sectores de arranque). En general, utilizan a estos huéspedes para poder transmitirse y ejecutarse inadvertidamente cuando lo hace el huésped.
- Son capaces de hacer copias (posiblemente modificadas) de sí mismos. Esta propiedad hace que su propagación sea exponencial.
- Se ejecutan automáticamente sin que el usuario sea, generalmente, consciente de ello.

Existen algunos programas que no cumplen todas estas características, pero que se suelen considerar como virus. Son los *gusanos* y los *troyanos*. Veamos en qué consisten.

Gusanos

Son programas autónomos. No necesitan un huésped, pero sí que son capaces de realizar copias de sí mismos, y de realizar acciones maliciosas y/o destructivas. En general, este tipo de virus se aprovecha de las rápidas redes de comunicación existentes, llegando a un ordenador y difundiéndose rápidamente a todos los ordenadores posibles desde el inicial. Es por esto que este tipo de virus se reproduce y se extiende a gran velocidad.

Típicamente, se difunden como ficheros adjuntos a través de correo electrónico. Una vez llegan a un ordenador, y son ejecutados se reenvían a todos los usuarios que aparecen en la libreta de direcciones de la aplicación empleada para gestionar el correo (MS Outlook, por ejemplo).

MyDoom: un ejemplo de gusano

En enero de 2004 apareció en la red *MyDoom*. Se trataba de un gusano que se transmitía como fichero adjunto en un correo electrónico. Al abrir (manualmente) el fichero adjunto, se ejecutaba el virus que procedía a instalarse en el ordenador.

Entre otras acciones este gusano:

- Enviaba copias de sí mismo a través del correo electrónico.
- Se copiaba en el directorio de ficheros compartidos del cliente P2P *Kazaa*.
- Abría un puerto para recibir conexiones TCP. Mediante estas conexiones podía recibir cualquier fichero binario y ejecutarlo.
- Finalmente, a partir del 1 de febrero inició un ataque por denegación de servicio contra la página web de la compañía SCO. Este tipo de ataques consiste en realizar un número tan grande de peticiones de conexión que el servidor se colapsa.

En sus primeras 36 horas de difusión, este gusano generó un tráfico de cerca de 100 millones de mensajes. En su llegada a España, supuso cerca del 30 % del tráfico de mensajes en la red en las primeras 24 horas.

Finalmente, se realizó el ataque sobre el servidor de SCO, colapsándolo. Hubo que cerrar el sitio y abrirlo de nuevo en una dirección web distinta.

Aparte de los daños que puedan producir cuando son ejecutados, uno de sus mayores peligros es que el mayor tráfico que generan ocupa ancho de banda de la red, llegando en algunos casos a colapsar algunas redes y servidores.

Troyanos

Los *caballos de Troya* no se consideran virus al no hacer (usualmente) copias de sí mismos. Son programas que forman parte de otro o bien “se hacen pasar por otro” y añaden una funcionalidad indeseada. En apariencia son benignos, pero esconden acciones de tipo malicioso.

También se consideran troyanos aquellos programas que están ocultos en el interior de otros (el huésped) aparentemente inofensivos. Cuando dichos programas son ejecutados, los troyanos se lanzan para realizar acciones que el usuario no desea. El usuario los instala en su sistema sin ser consciente del peligro (le están “tendiendo una trampa”), de ahí lo de Troya...

El principal medio de infección suele ser la ejecución de programas descargados a través de *redes de pares* (P2P). Sus efectos pueden llegar a ser muy peligrosos ya que facilitan ataques desde otros ordenadores o intrusiones, (mediante “backdoors” o puertas traseras), o pueden enviar información sobre la actividad de nuestro ordenador, e incluso permitir que el ordenador pueda ser controlado de forma remota.

¡Cuidado con los troyanos!

Instalar software en nuestro sistema de manera indiscriminada puede tener graves consecuencias.

Esto es especialmente peligroso en el caso de Windows dada la gran proliferación de programas en Internet y las capacidades de "ejecución de programas" (aprovechando la ignorancia del usuario) que tienen herramientas como *Internet Explorer* o *Outlook*.

En el caso de Linux éste es uno de los pocos peligros potenciales que se conocen (aparte de los *"exploits"*).

Para instalar software en el sistema es necesario hacerlo como `root`. Este hecho puede ser aprovechado por un troyano camuflado para instalarse en nuestro Linux con privilegios de `root` y causar graves perjuicios en nuestro sistema.

Solución: instalar software de fuentes fiables. Hay que evitar a toda costa instalar software descargado a través de redes P2P.

Además, en Linux los paquetes vienen "firmados digitalmente", garantizando su integridad. Los programas de gestión de paquetes rehúsan instalarlos cuando la firma no coincide con sus contenidos.

Medios de transmisión

Los virus, por su propia naturaleza, necesitan propagarse de un equipo a otro para poder llevar a cabo su cometido. Por ello, los creadores de virus los han hecho capaces de aprovecharse de cualquier medio de intercambio de datos. Sin embargo, la popularización de Internet en los últimos años la ha convertido en su hábitat preferido.

En general, los virus suelen viajar junto a sus huéspedes. Los huéspedes más habituales son los programas ejecutables. Así que cualquier medio por el que un programa ejecutable, y en general cualquier fichero, sea transmitido entre dos ordenadores, será susceptible de ser usado por los virus. Algunos medios son evidentes, como Internet, disquetes, discos duros extraíbles, memorias flash, etcétera, pero otros no lo son tanto. Es el caso, por ejemplo, de los discos compartidos por red, especialmente en redes de ordenadores con Windows (servicio SMB). Muchos de los virus actuales suelen detectar estos recursos compartidos, y copiarse en ellos para así expandirse más rápidamente por una red local Windows.

Las medios más habituales por los cuales nuestro ordenador puede ser infectado son éstos:

- Las páginas web pueden contener elementos que permiten la ejecución de programas. Si esto se combina con el uso de navegadores permisivos o mal diseñados, como es el caso del popular Internet Explorer, es posible que se ejecuten programas de forma inadvertida tan sólo accediendo a la página.
- Los mensajes de correo electrónico permiten la rápida difusión de virus. Se suelen trasmitir como ficheros adjuntos y la infección se produce con tan sólo abrir o leer los adjuntos (o incluso al acceder a los mensajes).

- Los programas ejecutables. En general, cualquier programa ejecutable puede ser el huésped de un virus, y éste se ejecuta al ejecutar el huésped. Por lo tanto, hemos de prestar atención a las fuentes de los que se han obtenido los programas y desconfiar de aquéllos descargados de páginas poco fiables u obtenidos mediante clientes P2P como *e-mule*, *Kazaa* y otros.

- Los ficheros con macros. Las macros son pequeños programas que ayudan a realizar/automatizar ciertas tareas y están incorporados dentro de documentos Word (ficheros con extensión .doc), hojas de cálculo Excel (extensión .xls) o presentaciones PowerPoint (extensión .ppt o .pps). También los documentos de OpenOffice permiten ejecutar macros. Algunos virus son capaces de modificar estas macros e insertar su código en ellas. Por lo general, las acciones de estos virus sólo afectan a las aplicaciones concretas de las que se aprovechan, aunque no por ello hay que ser menos desconfiado.

- La memoria del ordenador. Los virus permanecen ocultos en ella hasta que algún evento los active. Normalmente, un virus sólo llega a la memoria de un ordenador tras haberlo infectado. Pero una vez en memoria son capaces de infectar todos los programas y archivos que haya en ese momento en el ordenador.

- El sector de arranque. En el sector de arranque de un disco se aloja un programa que inicia la carga del sistema operativo. Un virus puede modificar este programa para cargarse en la memoria del ordenador, capturar interrupciones ("bloqueando" antivirus) y realizar sus actos malignos.

  Este medio es particularmente peligroso, en especial cuando afecta al sector de arranque de un medio extraíble (disquete, CD-ROM, DVD-ROM), ya que puede dar lugar a la contaminación de otros equipos.

  Hoy en día, este tipo de amenaza es escasa, ya que los SS. OO. son muy cuidadosos con las operaciones que pueden alterar los sectores de arranque, en particular del disco duro principal. Es necesario ser administrador para realizar la operación y suelen pedir confirmación para llevarla a cabo.

Curiosidades sobre los virus en macros

Son un fenómeno relativamente reciente. Suelen ir adjuntos a documentos Word o bases de datos como Access.

Se basan en la posibilidad que ofrecen algunas aplicaciones de extender su funcionalidad mediante la ejecución de otros programas (denominados macros) al cargar ficheros. Esto se debe al hecho de que estas aplicaciones incluyen lenguajes de programación especiales, los cuales están diseñados para "ayudar" a los usuarios a automatizar ciertas tareas relacionadas con el funcionamiento de la propia aplicación.

Un ejemplo son todas las aplicaciones del Office de Microsoft (y también de OpenOffice). Estos virus suelen infectar a los documentos de la aplicación.

continúa en la página siguiente . . .

. . . viene de la página anterior

Afectan a programas como Word, Excel, MS Outlook, etc. Pueden afectar a sistemas operativos distintos, ya que estas aplicaciones suelen ser multiplataforma (existe Word para Windows y para Mac).

El concepto de virus en macro funciona porque estos lenguajes de programación proporcionan acceso a la memoria y a los discos duros. Eso, de hecho, es lo que hacen otras tecnologías recientes, incluyendo los controles ActiveX y los applets Java.

Un virus en macro es simplemente una macro para uno de estos programas (y se hicieron famosos, precisamente, por infectar Microsoft Word). La mayoría de estos virus son "AutoOpen", es decir que se ejecutan causando el daño correspondiente cuando un documento o plantilla que contiene el virus se abre utilizando la aplicación en cuestión.

Uno de estos virus, *I love you*, se hizo famoso por colapsar diversas redes informáticas de empresas, corporaciones y entidades públicas en Europa y Estados Unidos.

La macro que contenía el virus era un script en Visual Basic Script que se ejecutaba al abrir el mensaje de correo electrónico que lo adjuntaba con Microsoft Outlook.

El virus (gusano) provocaba una sobrecarga de las redes informáticas al enviarse a todas aquellas direcciones que tuviese el usuario en la agenda de Outlook y borraba ficheros de ciertos tipos.

Además, creaba dos entradas en el registro principal de Windows, provocando su ejecución automática al arrancar dicho sistema operativo. Otro ejemplo, *Melissa*, era una macro de Word y tenía que ser abierto con dicha aplicación.

Efectos

Los efectos de un virus pueden variar de inocuos a muy graves, dependiendo de la mala fe del autor:

- Algunos destruyen información del sistema infectado, borrando ficheros tanto de datos como de configuración.
- Otros consumen recursos computacionales hasta inutilizar el sistema.
- Los hay que facilitan a terceros un uso indeseado del ordenador infectado. Esto lo hacen mediante el uso de *backdoors* o "puertas traseras". Es decir, pequeños programas que facilitan que un usuario desde el exterior del ordenador, acceda a él y ejecute otros programas, tomando, en ocasiones, el control total de la máquina.
- Los hay que transmiten información privada del propietario o características del equipo. A este tipo de software se le denomina *spyware*.

Medidas de protección

La primera medida de protección ante los virus es instalarse un buen programa antivirus. Los antivirus contienen una gran base de datos con "recetas" para detectar

virus y erradicarlos. Veamos cómo funciona un programa antivirus. Los programas antivirus se dividen en dos partes:

(1) El "motor" de *búsqueda* (programa antivirus en sí). Proporciona desde la interfaz con el usuario hasta el sistema interno de búsqueda.

(2) La base de datos que contiene las *descripciones* de los virus, nombres, características, patrones o secuencias de bytes, etc.

El funcionamiento del antivirus consiste en la coordinación entre las dos partes anteriormente mencionadas. El motor de búsqueda es el encargado de buscar en los discos y memoria aquellos patrones o secuencias identificados en la base de datos. Hay que tener en cuenta que por cada nueva actualización del motor de búsqueda, aparecen numerosas versiones de la base de datos de virus. De hecho, la base de datos *se ha de actualizar frecuentemente*: los creadores de virus diseñan nuevos virus casi a diario (bien partiendo de cero, bien mutando uno existente).

Algunos antivirus actúan preventivamente e impiden las infecciones antes de que se produzcan.

- Scanners: estos antivirus exploran el ordenador al arrancar para comprobar que no hay infección. Son programas que buscan secuencias de bytes o patrones característicos de los virus. Si se utilizan varios, hay que reiniciar la máquina entre uno y otro para evitar falsos positivos. Suelen comprobar que no han sido modificados y que no hay virus en memoria antes de ejecutarse. Es útil tener una copia en un disquete protegido contra escritura.
- Monitores: se quedan residentes en memoria y permiten analizar todas aquellas entradas de datos que el S.O. está gestionando. Tienen varias formas de acción:
  - Detectan y evitan "actividades sospechosas", tales como escrituras en el sector de arranque o en ficheros ejecutables.
  - Exploran los ficheros antes de que se vayan a ejecutar.
  - Añaden "firmas" a los ficheros para comprobar que no han sido dañados.
  - Tienen el inconveniente (cada vez menor) de ralentizar la máquina.
  - Muchos monitores tiene la capacidad de escanear automáticamente ficheros descargados y correo entrante.
- Comprobadores de integridad: obtienen una firma de cada fichero ejecutable de modo que si se modifica, el cambio sea detectable.

Algunos virus se aprovechan de fallos del S.O. y/o de las aplicaciones (ver *exploits* en la sección 7.2.2). Es importante, pues, actualizar estos elementos.

Prevención de infecciones

Hay que tener en cuenta que es *imposible* estar prevenido contra cualquier posibilidad de ser atacado por un virus. En cualquier caso, se pueden tomar una serie de medidas:

Elegir uno (o mejor varios) “scanners” o antivirus.

Acostumbrarse a pasar el “scanner” a todo programa que recibamos.

Crear un “disco de emergencia”.

Utilizar un programa residente de protección (monitorización).

Utilizar comprobadores de integridad.

Un virus es un programa que debe ejecutarse para estar activo. Por tanto un virus anexo a un mensaje de correo no hace nada hasta que se ejecuta. Conviene desactivar toda posibilidad de ejecución de programas vía correo-e y usar herramientas seguras para gestión de correo.

¿Qué hacer ante una infección?

Ante todo, mucha calma. En general, lo mejor es arrancar desde el disco de emergencia.

Hay que utilizar algún antivirus y buscar información al respecto. Las páginas web de los antivirus suelen contar con extensas bases de datos en las que se describe detalladamente cada virus, y la forma de erradicarlos, así como pequeños programas para hacerlo.

Formatear el disco es la última opción y casi nunca es necesario. Si se infecta el sector de arranque, generalmente se puede restituir con `FDISK /MBR`.

Caso de haber algún programa infectado, si podemos recuperarlo de alguna copia de seguridad o volver a instalarlo, es mejor que intentar limpiarlo.

Disco de emergencia

Hay que acostumbrarse a tener un disco “limpio” que pueda arrancar el ordenador con una serie de herramientas básicas.

Los antivirus suelen incluir la opción de crear un disco de arranque.

Debe estar protegido contra escritura para evitar infecciones en su sector de arranque. Si esto ocurriese el disco sería inutilizable.

Algunos antivirus

La mayoría de los antivirus son de pago, aunque muchos de ellos ofrecen versiones libres, pero sin toda su funcionalidad. He aquí algunos de ellos:

- *AVG Antivirus* distribuye una versión gratuita.
  `http://free.grisoft.com`
- *Panda Software Antivirus*
  `http://www.pandasoftware.com`
  Dispone también de un scanner gratuito *online*, aunque sólo puede ejecutarse desde el navegador *Internet Explorer* o *Mozilla Firefox*.
  `http://www.pandasoftware.com/activescan`
- *McAfee VirusScan*
  `http://es.mcafee.com`
- *Norton Antivirus*
  `http://www.symantec.com/nav/nav_9xnt/`
- Y muchos más…

También es una medida importante adoptar una actitud de prevención y constante alerta en los hábitos de uso del ordenador. Algunas consejos en este sentido son éstos:

- No usar algunas aplicaciones poco fiables: *Internet Explorer*, *Outlook* y *Outlook Express*. Es mejor emplear en su lugar otras como *Mozilla Firefox* y *Mozilla Thunderbird*. En caso de seguir usando esas aplicaciones, se debe mantenerlas constantemente actualizadas.
- No abrir correos (y, sobre todo, no abrir los ficheros adjuntos) de origen incierto, o de aspecto sospechoso (con el asunto en inglés, por ejemplo) aunque provengan de un conocido.
- Al abrir algunas páginas web, se abren ventanas emergentes que nos piden autorización para instalar programas. No hay que aceptar nunca.
- Mucho cuidado con los programas descargados mediante clientes P2P. A ser posible, no hay que descargarlos nunca. Y si los descargáis, escaneadlos inmediatamente con un antivirus actualizado.

Finalmente, debes tener en cuenta que muchos virus… ¡ni siquiera existen!. Es frecuente encontrar y recibir por correo electrónico falsas alarmas y bulos de todo tipo (*hoax*): *"El virus tal y cual ha infectado los ordenadores que corren tal o cual sistema operativo. Si ud. encuentra un fichero denominado CONFIG.SYS en su ordenador, ha sido infectado. Para eliminar el virus, borre el fichero. Distribuya este aviso."*

Habitualmente, estos mensajes indican que la presencia de un virus se detecta por la presencia de un fichero concreto (como en el ejemplo anterior) y os piden que lo borréis: no lo hagáis. Normalmente, se trata de un fichero importante. Tampoco reenviéis el mensaje.

El objetivo de estos bulos no sólo es difundir información falsa, sino desperdiciar recursos y llenar los buzones de correo de los usuarios al pedir su reenvío a todos los contactos de quien recibe el mensaje. También pueden inducir al usuario ignorante a cometer acciones que pueden dañar su sistema de forma, a veces, irreparable.

¡Cuidado con los bulos!

Otra vertiente de los virus son las noticias que se difunden acerca de ellos y que resultan ser *bromas pesadas* (hoax), que suelen causar más daño que muchos virus.

El objetivo de estos mensajes de correo electrónico no sólo es difundir información falsa, sino desperdiciar gran cantidad de ancho de banda y llenar los buzones de correo de los usuarios al pedir su reenvío a todos los contactos de quien recibe el mensaje.

También pueden inducir al usuario ignorante (lego en informática) a cometer acciones que pueden dañar su sistema de forma, a veces, irreparable (por ejemplo, diciendo que borre determinados ficheros de configuración, cuya "misión" en el sistema el usuario desconoce).

Ejemplo: el virus *Good Times*, el cual supuestamente se transmitía vía correo electrónico. Se decía que este virus se activaba al leer un mensaje que contuviese la frase "Good Times" en su asunto, borrando el disco duro o destruyendo el procesador.

Por supuesto, esto era una mentira, pero muchos lo creyeron y empezaron a hacer circular el mensaje por Internet. Desde entonces, han aparecido muchas versiones de este mensaje.

Una de ellas asegura que el virus podría destrozar el procesador al "ponerlo en un salto binario infinito de extrema complejidad"; frase sin sentido pero capaz de aterrorizar a legos en materia de computación. El pánico puede llegar a ser más destructivo que cualquier virus.

Algunos de los "bulos" más conocidos son: *Win a holiday*, *Irina*, *Penpal Greetings*. . . En `http://www.symantec.com/avcenter/hoax.html` se puede encontrar una buena lista de estos bulos.

7.2.2. Intrusiones

Otro importante riesgo que corre cualquier ordenador conectado a una red es el de las intrusiones. Entendemos por intrusión el acceso no autorizado de un usuario a una máquina concreta sin que el propietario de ésta sea consciente de ello.

Estas intrusiones suelen realizarse a través de fallos ("bugs") o "agujeros" en el S.O. o en las aplicaciones que proporcionan servicios de comunicación. Los *exploits* son programas que se dedican a explotar (de ahí su nombre) fallos en el diseño de esta aplicaciones críticas e incluso del propio S.O.

Estos fallos de programación y/o diseño suelen producirse debido a situaciones no contempladas a la hora de diseñar los programas. Por ejemplo, algunos ocurren cuando los programas no controlan la entrada de datos o cuando ésta excede de un número determinado de caracteres (*buffer overflow*). Algunos fallos afectan a cosas tan importantes como servidores WWW, FTP, de correo, etc. y pueden causar problemas importantes como la negación de un servicio, el bloqueo de una máquina o incluso accesos no autorizados con privilegios de administrador del sistema.

Otra idea importante relacionada con las intrusiones es el concepto de *puerto*. Como ya sabemos, un PC necesita tener una dirección electrónica para poder identificarse y comunicarse con el resto de ordenadores conectados a Internet. Esa dirección electrónica es la *dirección IP*. Pero esto no es suficiente, ya que en Internet se pueden utilizar muchos y diversos servicios y es necesario poder diferenciarlos. Como ya vimos, la forma de hacerlo es mediante *el mecanismo de puertos TCP*.

Las puertos son identificadores especiales que, para cada ordenador, pueden identificar una multitud de servicios. Así, por ejemplo, cuando pretendemos acceder al servidor web de una determinada máquina, lo haremos dando su número de IP, e indicando el puerto TCP 80, que es el que usan los servidores web. De esta forma, es posible tener múltiples aplicaciones enviando y recibiendo desde un único equipo sin que haya confusiones.

Existen más de 65.000 puertos TCP diferentes usados para las conexiones de red. Si en tu ordenador se cuela, a través del correo electrónico por ejemplo, un virus capaz de abrir alguno de estos puertos, el resultado es que la puerta de tu casa quedará abierta. Toda una golosina para los intrusos. Además, puedes estar seguro de que cualquier puerto abierto que tú no controles (en ocasiones es posible que ni tan siquiera sepas que existe) es una invitación para que puedan fisgar en tu equipo, robarte información confidencial y ocasionarte multitud de problemas.

En resumen, las principales vías para introducirse en tu ordenador son:

- Usar un *exploit*, es decir, provocar un fallo en una determinada aplicación. Una vez provocado el fallo es posible tomar el control de la máquina.
- Abrir un *backdoor*, es decir, abrir un puerto TCP para permitir el acceso desde el exterior. Esta es una acción típica de la mayoría de los virus, troyanos, y *exploits*.
- Hacer un escaneo de puertos (*port scan*): comprobar todos los puertos con el fin de encontrar alguno desatendido. En muchos casos, estos puertos están atendidos por aplicaciones con "bugs" conocidos. Una vez identificada la aplicación tan sólo hay que usar el *exploit* adecuado y ¡ya están dentro!

Por si fuera poco, cualquiera puede conseguir un *exploit* y utilizarlo contra nuestra máquina sin ni siquiera saber cómo funciona y sin unos conocimientos mínimos. Con un poco de cuidado y "lectura" (saber lo que hacemos en todo momento) podemos evitar fácilmente que cualquier persona sea capaz de atacar con éxito un equipo simplemente por haber visto una película, consultado un par de páginas web y descargado y ejecutado un programa que ni ha hecho ni entiende.

Protección frente a intrusiones

Algunos consejos para mantenernos protegidos frente a intrusiones son éstos (aplicables tanto en Windows como en Linux):

- Mantener actualizadas todas las aplicaciones que hagan uso de la red, en particular los servidores (ftp, telnet, openSSH, web server, etcétera). Ya explicamos al principio de la sección 7.2 cómo es posible hacer esto.
- Si no son necesarios, desactivar dichos servicios. En general, cuando instalas una distribución de Linux, estos servicios están inactivos. En Windows no suelen ni estar instalados.
- Instalar un cortafuegos o *firewall*. Un cortafuegos es una aplicación que controla el acceso y uso de los puertos TCP. Las principales actividades que desarrolla un cortafuegos son éstas:
  - Bloquea los puertos que no se vayan a utilizar. De esta forma, sabemos en todo momento qué puertos se están usando y para qué.
  - Pide autorización antes de que un programa acceda a determinados servicios de Internet, es decir, de que envíe mensajes desde el equipo. De esta forma, sabemos qué programas se están comunicando con el exterior desde nuestro equipo, pudiéndoles impedir el acceso si no nos parece conveniente.
  - Limita el acceso desde otros equipos, impidiéndolo si fuese necesario. Así, sabemos desde qué equipos se están produciendo comunicaciones hacia el nuestro.
  - En todo caso, monitoriza y registra las actividades en red de nuestro equipo y, en particular, aquellas actividades de mayor riesgo.

Existen diversos cortafuegos disponibles para sistemas de la familia Windows: *ZoneAlarm* (versión no profesional) es uno bastante completo y de uso gratuito. También Windows XP dispone de un cortafuegos (que se instala y activa por defecto con el *Service Pack 2*). En general, hoy en día, los cortafuegos ya son reconocidos como herramientas básicas de seguridad para ordenadores domésticos y las mismas compañías que ofrecen antivirus ofrecen también cortafuegos.

En el caso de distribuciones Linux, el cortafuegos viene ya integrado en el *kernel*[2]. Habitualmente, se gestiona mediante la utilidad *iptables*, también llamada *netfilter* según sistemas. Con esta utilidad se pueden especificar toda clase de reglas mediante instrucciones expresadas en ficheros de texto. Este método no es demasiado sencillo, por lo que existen diversos clientes gráficos para usar la utilidad (*knetfilter* o *firestarter*, por ejemplo) aunque aún requieren cierto conocimiento sobre la sintaxis de las reglas. Por ello, es habitual que casi todas las distribuciones dispongan de herramientas propias que simplifiquen las configuración del cortafuegos. En el caso de SuSE, por ejemplo, este gestor aparece en el apartado de *Seguridad y Privacidad* del *Centro de Control*.

[2]Núcleo del S.O.

7.2.3. Otros problemas

Existen una serie de problemas que, aunque no suponen un riesgo para la seguridad y el buen funcionamiento de los equipos, sí que suponen serias molestias para los usuarios y navegantes de Internet ("internautas").

SPAM

Por SPAM entendemos los mensajes de correo-e que, generalmente con fines comerciales, son enviados de forma indiscriminada a usuarios que no lo han solicitado (¡ojo!, muchas veces son timos). Se trata de una fuente de molestias y, en muchos casos, puede incluso llegar a colapsar un buzón de correo.

Una excelente herramienta para interceptar el SPAM es *SpamAssassin*. Usa técnicas basadas en reglas bayesianas para aprender a identificar el correo no deseado. En general, este tipo de herramientas se utilizan en los servidores de correo. Por ejemplo, el servidor de correo de la UJI utiliza estas herramientas, por lo que todo el correo que llega a nuestras cuentas ha sido filtrado previamente en busca de SPAM (y también de virus). Además, la mayoría de los clientes de correo suelen disponer de filtros que permiten eliminar parte del SPAM si son configurados de forma adecuada.

Igualmente, existen algunas reglas de conducta que pueden ayudar a disminuir esta molestia. Para empezar, debemos ser especialmente cuidadosos en difundir nuestra dirección de correo, y no darla alegremente en el primer formulario que aparezca mientras navegamos. También, resulta habitual que nos lleguen correos publicitarios periódicamente, los cuales solicitan el envío de un mensaje a una dirección de respuesta para evitar la recepción de futuros mensajes por el estilo. En esos casos, al responder, lo que hacemos es confirmar que la dirección de correo es válida y que la leemos de vez en cuando, lo cual es una incitación para que nos envíen más correo. De todas formas, si se navega de forma habitual, es inevitable que nuestra dirección de correo acabe en manos poco escrupulosas.

Nosotros también podemos constituir una molestia

Es habitual recibir mensajes que piden ser reenviados al máximo número de contactos posibles, ya bien sea por interés humanitario o de otro tipo.

¡Y quien no ha recibido y reenviado esos chistes tan graciosos! Aparte de que muchos de los mensajes "humanitarios" que se reciben no son ciertos, hay que ser consciente de a quién reenviamos esos mensajes, y si es apropiado hacerlo. No a todo el mundo le apetece recibirlos y, en muchos casos, también pueden ser una molestia.

Spyware

Se trata de software que, de forma encubierta, utiliza nuestra conexión a Internet para extraer y trasmitir datos e información de nuestro ordenador o sobre nuestro comportamiento. Por supuesto, sin nuestro consentimiento.

Al contrario que los virus, no pretende provocar daños en nuestro equipo, sino que

tan sólo pretende "espiar" nuestras actividades, pero aún así es una agresión a nuestra privacidad. Además, en la mayoría de las ocasiones va integrado en aplicaciones legales y suele pasar desapercibido. Ha sido puesto allí por los creadores de éstas de forma intencionada.

En general, al no ser un virus, los antivirus no suelen detectarlos ya que no han sido preparados para ello. Sin embargo, las últimas versiones de antivirus ya suelen integrar funcionalidad especializada en *spyware*.

Microsoft permite descargar de forma gratuita (siempre que se disponga de una licencia válida de XP o Vista) una herramienta muy interesante para ayudar al control del *software* instalado en nuestro equipo: *Windows Defender*. Es un programa que escanea periódicamente el sistema en busca de *spyware* y otros programas malintencionados. Si se instala, conviene actualizarlo periódicamente (si hemos activado las actualizaciones automáticas de Windows, esto se hará automáticamente, valga la redundancia).

Dialers

Se trata de programas encargados de marcar números de teléfono para el acceso a un servicio vía una conexión mediante módem telefónico (RTC, Red Telefónica Convencional). Usados de forma maliciosa, pueden suponer una gran fuente de ingresos (para el receptor de la llamada, por supuesto). Esto es así ya que las llamadas se realizan a números con una tarificación especialmente onerosa. Algunas páginas web están diseñadas para instalarlos de forma automática e inadvertida.

La mejor forma de controlarlos es instalar un programa que supervise y autorice las llamadas que se realicen desde nuestro equipo. Y por supuesto, si en alguna página se pide autorización para instalar algún programa, no autorizarla nunca.

Phishing

En general, se refiere al acto de adquirir, de forma fraudulenta y a través de engaño, información personal (contraseñas, detalles de tarjeta de crédito, etc.), haciéndose pasar por alguien digno de confianza con una necesidad verdadera de tal información en:

- un correo-e con "sello oficial",
- mensaje instantáneo,
- cualquier otra forma de comunicación personal.

Originalmente, se consideraba como *phishing* al acto que consistía en recomendar la visita a una página web falsa, haciendo creer al visitante que se encontraba en la página original (mediante una copia).

La vía de difusión más habitual de esta técnica es el correo electrónico, aunque últimamente se han detectado vías alternativas como el teléfono o el fax. Normalmente se utiliza con fines delictivos, duplicando páginas web de entidades financieras de renombre. Una vez en las páginas falsas, se pide al visitante que introduzca datos personales (claves de acceso, etc.), que posteriormente son usados por los creadores de la estafa.

Las entidades recomiendan que no se revelen nunca las claves personales aunque sean pedidas en su nombre.

Se dice que el término “*phishing*” proviene de *password harvesting fishing* (cosecha y pesca de contraseñas).

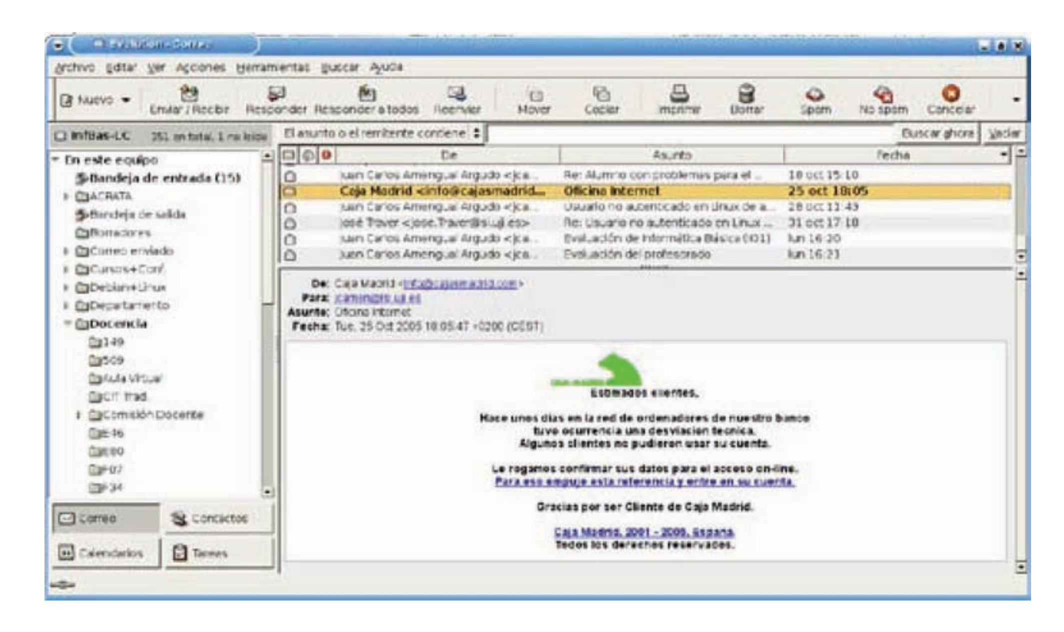


Figura 7.6. Un ejemplo de *phishing*: correo-e falso para “pillar” los datos de una cuenta bancaria

En la Figura 7.6 podemos apreciar un ejemplo de este tipo de “actuación”. Observa atentamente que el correo electrónico enviado proviene de `Caja Madrid <info@cajasmadrid.com>`. El dominio es un tanto “sospechoso”. Se puede averiguar más cosas con la orden `whois` de Linux. Para ello, hemos de pasarle su dirección IP (averiguada con la orden `host`) e indicarle la base de datos donde buscar:

- `whois.ripe.net` para Europa,
- `whois.arin.net` para América,
- `whois.apnic.net` para Asia.

La dirección de la web que apunta en el mensaje es `http://www.cajamadriid.com`, lo cual indica bien a las claras que se trata de un *fake* (engaño). Veamos la información que muestra `whois`:

```
$ host cajasmadrid.com
cajasmadrid.com has address 82.152.107.201
cajasmadrid.com mail is handled by 20 mail.cajasmadrid.com.
$ whois -h whois.ripe.net 82.152.107.201
% This is the RIPE Whois query server #3.
% The objects are in RPSL format.
...
% Information related to '82.152.0.0 - 82.153.255.255'

inetnum:        82.152.0.0 - 82.153.255.255
org:            ORG-EN5-RIPE
```

```
netname:        UK-ECLIPSE-20031110
descr:          Eclipse Networking Limited
country:        GB
...
phone:          +44 1392 202345
fax-no:         +44 1392 202346
e-mail:         hostmaster@eclipse.net.uk
...
person:       Mark Lang
address:      c/o Eclipse Internet,
address:      Portland House, Longbrook Street,
address:      Exeter, Devon EX4 6AB
address:      GB
phone:        +44 1392 333309
fax-no:       +44 1392 333319
...
```

En contraposición, analicemos la información devuelta para el *dominio auténtico*:

```
$ host cajamadrid.com
cajamadrid.com has address 213.164.164.74
$ whois -h whois.ripe.net 213.164.164.74
% This is the RIPE Whois query server #3.
% The objects are in RPSL format.
...
% Information related to '213.164.164.0 - 213.164.164.254'

inetnum:      213.164.164.0 - 213.164.164.254
netname:      Cajamadrid
descr:        Caja Madrid Internet Public Services
country:      ES
...
role:         Caja Madrid Network Dept.
address:      C/Gabriel Garcia Marquez, 1
address:      28230 Las Rozas - Madrid
e-mail:       aruizori@cajamadrid.es
...
person:       Susana Munoz Gomez
address:      Paseo de la Castellana, 189, Planta 18
address:      Madrid
address:      E-28046
address:      MADRID
address:      SPAIN
phone:        +34 914235102
fax-no:       +34 914235833
e-mail:       smunozgo@cajamadrid.es
...
```

Es una forma de ataque de la *ingeniería social* (los timos de toda la vida) y nos enseña, una vez más, que no hay que fiarse de nadie y ser muy precavidos a la hora de

enviar datos personales por Internet.

Debes saber que, en el campo de la seguridad informática, *ingeniería social* es la práctica de obtener información confidencial a través de la manipulación de usuarios legítimos. Un ingeniero social usará medios electrónicos para engañar a la gente y llevarla a revelar información sensible, o bien a violar las políticas de seguridad típicas. Con este método, los ingenieros sociales aprovechan la tendencia natural de la gente a confiar en su palabra, antes que aprovechar agujeros de seguridad en los sistemas informáticos. El principio por el que se rige la ingeniería social es el de que "los usuarios son el eslabón débil".

Más referencias sobre seguridad

En Internet se pueden hallar múltiples referencias para profundizar en temas de seguridad. En particular, una que ha sido usada profusamente para la redacción de este capítulo es `http://www.seguridadenlared.org`. También resulta muy interesante echarle un vistazo a `http://alerta-antivirus.red.es/portada/`

7.2.4. Copias de seguridad

Cada vez más, utilizamos los ordenadores para un mayor número de actividades y, con el tiempo, vamos depositando en ellos mayor cantidad de datos y documentos, que nos ha costado tiempo y trabajo crear. Se trata de datos *valiosos*. Estos datos están almacenados en dispositivos que consideramos fiables[3], pero en los cuales no podemos confiar eternamente, porque:

- éstos pueden fallar por averías,
- los programas que manejan los datos pueden tener errores y corromperlos,
- pueden ser atacados por virus o usuarios "maliciosos",
- puede irse la luz en medio de una operación de escritura de datos,
- el usuario puede destruir involuntariamente datos,
- etc.

Hay dos tipos de usuarios: los que han perdido datos valiosos y los que están a punto de perderlos.

Es necesario tener una política de realización de copias de seguridad (*backups*) de los datos en otros dispositivos (cinta magnética, discos magneto-ópticos, discos duros, DVD, etc.) para evitar pérdidas irreparables de datos debido a causas como las que acabamos de comentar.

[3]Los discos duros, por ejemplo.

Tipos de copia de seguridad

Hay varios tipos de copia de seguridad:

Copias de seguridad globales: guardan una imagen de todo lo que hay en el disco o unidad. Con los tamaños de disco actuales es poco operativo si no se hace sobre una cinta o similar, puesto que se ha de disponer de un sistema de almacenamiento de igual o mayor capacidad (puede ser un poco menos si usamos herramientas de compresión[4]). Otro de los inconvenientes es la lentitud, es decir el elevado tiempo que tarda en realizarse la copia de seguridad.

Copias de seguridad parciales: se seleccionan aquellos ficheros que se quieren guardar, generalmente los que puedan resultar críticos tanto para el usuario como para el sistema (bases de datos, ficheros de configuración, documentos imprescindibles o que se requiere conservar, etc.). Como usuarios seleccionamos la carpeta o carpetas y la estructura de directorios que deseamos “respaldar”[5]. No necesitan, en principio, herramientas especiales o específicas.

Copias de seguridad incrementales: se hacen copiando únicamente aquellos ficheros cuyo contenido haya cambiado desde la última copia de seguridad. Normalmente, este tipo de copias de seguridad se emplean conjuntamente con la realización de copias globales: la política suele ser hacer una copia global al cabo de un periodo de tiempo (p.e. 1 mes) y hacer copias incrementales a intervalos más cortos de tiempo (p.e. cada tres días). Ello se debe a que las copias globales suelen llevar mucho más tiempo que las incrementales.

Otras copias de seguridad

Copias de seguridad temporales: consisten en la creación de copias de los ficheros en el propio disco, generalmente con la extensión `.BAK` y de forma automática. Por ejemplo, `Xemacs` suele crear un backup temporal de cada fichero al modificarlo, añadiéndole el carácter ~ al nombre del fichero modificado.

Copias de seguridad en serie: especialmente utilizadas al desarrollar programas. Consisten en ir haciendo copias de las distintas versiones que se van creando del fichero o ficheros implicados en un proyecto software. De esta manera se puede volver a versiones anteriores si es necesario. Suelen utilizarse conjuntamente con programas (CVS) que permiten la gestión (concurrente) de proyectos de programación por parte de grupos formados por desarrolladores diferentes (*repositorios* de programas).

[4] 7-zip, gzip.

[5] Las copias de seguridad o *backups* también se denominan *copias de respaldo*.

Herramientas

Existen muchos programas, más generales, cuyas características permiten emplearlos para realizar copias de seguridad de alguno de los tipos mencionados. Otros son programas específicos para copia de seguridad. Vamos a presentar algunos, agrupados por funcionalidad. Pero antes de abordar estos asuntos, quisiéramos tratar una cuestión de nomenclatura:

- Fichero: nos referimos por fichero a la entidad lógica básica para guardar una colección de informaciones relacionadas entre sí y definidas por su creador.
- Archivo: en este caso, nos referimos a una agrupación o colección que sirve de almacén de ficheros y/o directorios. Bajo este punto de vista, un archivo no es sinónimo de fichero.

Hecha la salvedad, vamos a presentar los programas agrupados por funcionalidad:

- Programas de copiado de ficheros.

  En principio, se limitan a crear un fichero cuyo contenido es una copia de otro. Algunos pueden hacer copias recursivas, es decir, obtener una copia de todos los ficheros y directorios que "cuelgan" de un directorio dado. También hay programas que permiten realizar estas copias de información en otras máquinas conectadas a una red de ordenadores (Internet).

  COPY: (MS-DOS, Windows) programa que se puede utilizar para hacer copias parciales cuando abarquen un número reducido de ficheros.

  XCOPY: (MS-DOS, Windows) con más posibilidades que COPY, como comprobación del atributo archivo, verificación de la copia, posibilidad de copiar subdirectorios y copia de los ficheros a partir de una fecha dada.

  cp: (Unix) incluye muchas de las funcionalidades de los anteriores y algunas más.

  ftp: (Unix, Windows) permite realizar transferencias de ficheros y/o directorios completos a otros ordenadores conectados a una red informática (y, por consiguiente, copias a través de la red). `sftp` funciona de manera muy similar, con la salvedad de que la información "viaja" cifrada por la red. En realidad es un servicio: hay muchos programas (incluso gráficos) capaces de hacer transferencias FTP.

  scp: (Unix, Windows) similar a `cp` con dos salvedades: permite transferir información remotamente por una red de ordenadores y la información se transmite cifrada. También es un servicio: existen programas gráficos, como *WINscp*, que permiten transferir ficheros de esta manera.

- Empaquetadores.

  Generan un único archivo en el que se ha empaquetado un conjunto de ficheros y/o la estructura de directorios que cuelga de una ruta dada. Algunos pueden, además, comprimir los datos para que ocupen menos espacio.

Tar: es un programa muy usado en el “mundo” Unix que permite empaquetar (y en algunas versiones comprimir) un conjunto de ficheros en un único archivo. Desarrolla una funcionalidad similar a `cpio` o a PKZIP (MS-DOS) o incluso *WinZip*, *Winrar* o *7-zip* (Windows). Es útil para hacer copias de seguridad y para intercambiar ficheros con otras personas.

cpio: programa Unix para manejar archivos compuestos de ficheros, esto es, archivos que contienen múltiples ficheros (como `tar`). Permite crear archivos, extraer ficheros, copiar ficheros de un sitio a otro... Maneja gran número de formatos de archivo (incluyendo el formato generado por `tar`).

afio: (Unix). Al igual que los anteriores, permite manejar archivos compuestos de ficheros (archivos AFIO). Los archivos AFIO son portables ya que la información de “cabecera” o control está en formato ASCII. El formato es como el de `cpio`. Sus ventajas: trata de mejor forma las posibles corrupciones en los datos de los ficheros (si hay un fichero corrupto no se echa a perder toda la copia de seguridad, sólo el fichero), soporta archivos multi-volumen, permite compresión de datos más robusta y segura que la de `tar` o `cpio`. Se usa comúnmente como el programa encargado de crear archivos (volúmenes de ficheros) en un sistema o programa de copias de seguridad.

- Programas específicos.

  Realizan las funciones de los anteriores con, posiblemente, interfaces mejorados y añaden ciertas funcionalidades: copias incrementales, gestión de versiones de ficheros, almacenamiento en cinta o disco, almacenamiento multivolumen (incluso en CD-RW o DVD-RW), etc.

interfaces: konserve, KDat, etc: en el “mundo” Linux existen programas que presentan interfaces más amigables (menús textuales en consola) y están pensados para hacer copias de seguridad con determinados dispositivos (cintas, discos o cintas SCSI, en red, etc.) como `KDat` o `amanda`. También existen interfaces gráficas para gestionar programas como `tar`, `cpio` o `afio`. En particular, se recomienda “echar un vistazo” a programas como `konserve` y `cdbackup` (copias parciales), `faubackup` (copias globales e incrementales), `Mondo Rescue` (¡excelente!, para copias globales) y `multicd` (copias globales y parciales).

Copia de seguridad de Windows (Windows Backup): permite que se pueda planificar las copias y hacerlas sobre un archivo o sobre dispositivos específicos.

Programas específicos: lo más aconsejable si se pretenden hacer de forma seria. En muchos casos vienen con las unidades de cinta. Dada la flexibilidad y características de Linux, es muy frecuente encontrarse programas o “scripts” de backup preparados específicamente por el administrador del sistema.

Copias de seguridad en casa

Llegados a este punto, ya debemos estar bastante concienciados de la necesidad de hacer algún tipo de copia de seguridad de nuestros datos. Ahora bien, ¿cómo podemos hacerlo en nuestros ordenadores domésticos? He aquí algunos aspectos a tener en cuenta:

- Las copias de seguridad han de realizarse *sobre un medio secundario*. Este deberá ser un medio distinto y, a ser posible, en un lugar distinto al original. Es decir, si queremos copiar los datos contenidos en un disco duro, esta copia se realizará sobre un medio distinto: otro disco duro (no son tan caros), un CD-ROM, un DVD-ROM, y si disponemos de más dinero, sobre una cinta magnética. Con esto evitamos que, si se dañan los datos originales, la copia se vea afectada.

  Cabe incluso la posibilidad de que, si disponemos de varios ordenadores en red, las copias de seguridad se realicen en medios remotos.

- Deben realizarse *periódicamente*. Ya que las copias globales suelen requerir mucho tiempo, se puede hacer una global periódicamente, y copias incrementales (más rápidas) sobre ella con una mayor frecuencia. Estas copias podrían realizarse en un disco duro secundario.

  Cada vez que se crea un nueva copia global se reemplaza la copia global e incrementales anteriores. En lugar de borrar simplemente estas copias anteriores, pueden ser almacenadas en algún otro medio, como un DVD regrabable, y así mantener estas copias durante mucho más tiempo.

  Relacionado con esto, cabe comentar que muchas empresas tienen políticas de seguridad muy estrictas que incluyen, por ejemplo, la necesidad de disponer de cámaras fuertes ignífugas para almacenar ciertas copias.

- Una buena idea para simplificar la realización de copias de seguridad es la de mantener juntos, en un sólo directorio o partición, todos los datos de los que queramos hacer copias. Así pues, no es mala idea tener una partición de un disco dedicada exclusivamente a almacenar los datos valiosos, dejando en otras particiones los binarios de los programas, y otros datos menos valiosos. De esta forma a la hora de hacer la copia de seguridad sólo hay que guardar los datos de esa partición, y no tener que ir buscando los datos dispersos por varios lugares.

- Hay que pensar bien de qué datos pretendemos mantener copias de seguridad. Dependiendo de la cantidad de datos, necesitaremos más o menos espacio en los medios auxiliares. Si sólo guardamos datos de trabajo, estos pueden ocupar relativamente poco espacio, y tal vez quepan en un DVD-ROM o un CD-ROM.

  En cambio, si lo que queremos es hacer una copia de respaldo de todo el sistema, puede ser necesario mucho más espacio. Este tipo de copias son útiles para recuperar el sistema completo tras un fallo catastrófico que lo haya destruido por completo. Con sólo restaurar la copia de seguridad, tendríamos rápidamente la instalación tal como la dejamos, en lugar de tener que instalar y configurar de nuevo todas las aplicaciones. Eso sí, este tipo de copias requieren mucho más espacio.

- En la sección 7.2.4 hicimos una somera descripción de las distintas herramientas disponibles para la realización de *backups*. En caso de que ninguna de las herramientas más avanzadas existentes satisfagan plenamente las necesidades, siempre es posible programar un *script* para la realización de copias de seguridad *ad hoc*.

7.2.5. Consejos y precauciones con el uso de contraseñas y cuentas

Con el uso habitual de Internet es común tener varias cuentas y varias contraseñas para acceder a ellas. En particular, la UJI proporciona una cuenta a todos los alumnos, y es necesaria una contraseña para acceder a ella. Es responsabilidad del usuario de la cuenta hacer un uso adecuado de ésta, y tomar ciertas precauciones.

- Es necesario elegir una buena contraseña: hay que evitar palabras que aparezcan en cualquier diccionario, nombres personales, "qwerty" o dejar la contraseña vacía. También es conveniente añadir dígitos a las contraseñas y mezclar mayúsculas y minúsculas.

  El servidor de autenticación usado en `anubis`, ya comprueba muchas de estas cosas, e incluso obliga a cambiar de contraseña de forma periódica.

- Hay que tener cierto cuidado con los permisos de los ficheros, directorios y programas. En general, es preferible que sean sólo accesibles por el usuario propietario.

- Cuando se abandona un ordenador en el que se ha estado trabajando, hay que cerrar la sesión. De lo contrario, el siguiente usuario tendría acceso a los datos de nuestra cuenta.

- Muchos navegadores tienen la "utilidad" de recordar usuario y contraseña cuando se accede a un sitio web. Hay que ser muy cuidadoso con este tema, ya que si se trata de un ordenador público (laboratorios o aula de acceso libre) que arranque Windows o Linux con algún usuario genérico de trabajo (no pide identificación) estos datos podrían quedar almacenados.

7.3. PIRATERÍA

La posibilidad de obtener copias de información digital virtualmente indistinguibles del original hace que sea atractivo copiar programas sin pagar por ellos, ya que el coste de la copia es, por regla general, ridículo. Y esto no es sólo aplicable a los programas de ordenador, si no que también es aplicable a cualquier medio audiovisual digital (películas en DVD, CD de audio, etc.). Las copias no autorizadas son ilegales; los programas están protegidos por la ley de la propiedad intelectual y la nueva reforma del código penal (Ley Orgánica 15/2003 en vigor desde el 1 de octubre de 2004). En este sentido, la ley 23/2006 (julio 2006) modifica el texto refundido de la Ley Orgánica sobre propiedad intelectual: básicamente, amplía el canon a una mayor variedad de

soportes digitales (reproductores MP3, por ejemplo), pero mantiene el derecho de *copia privada*; aunque en este caso se restringe a un ámbito "familiar" y se ha de usar un medio legal para efectuarla (con lo que el autor de estos apuntes entiende que se mantiene igual que hasta ahora).

Cómo elegir una contraseña

El principal problema al escoger una buena contraseña es que luego suele ser difícil recordarla. A continuación, describimos un sencillo mecanismo para crearlas y luego poder recordarlas fácilmente:

(a) Comenzamos con una frase fácil de memorizar (por ejemplo una cita):
"Los ordenadores son buenos siguiendo instrucciones, pero no leyendo la mente" (Donald Knuth)

(b) Nos quedamos con la primera letra de cada palabra:
"Losbsipnllm"

(c) Las letras correspondientes a sustantivos las ponemos en mayúsculas y el resto en minúsculas:
"lOsbsIpnllM"

(d) Sustituimos las palabras correspondientes a artículos "el, la los, la" por un "1":
"1OsbsIpnl1M"

De esta forma, aunque el método es mejorable, se puede generar una contraseña compleja a partir de un frase sencilla de memorizar.

Los piratas vulneran los derechos de autor copiando y distribuyendo (cuando no vendiendo) copias no autorizadas por el propietario. El autor tiene derecho a decidir si su obra puede copiarse y en qué condiciones. No obstante lo dicho en el párrafo anterior, conviene distinguir entre dos conceptos totalmente distintos:

- *Copia de seguridad*: referida *exclusivamente* al software. *No se necesita autorización* del autor para hacerla, ni se puede *impedir por contrato*. Consiste en reproducir (copiar) software para la utilización *legítima* del mismo por parte del usuario. El concepto "legítimo" es importante: *sólo puede hacer la copia de seguridad para su propia utilización el propietario legítimo del software*.

- *Copia privada*: referida a *obras* (literarias, musicales, audiovisuales, etc.). Las obras *ya divulgadas* pueden reproducirse sin autorización del autor *para uso privado del copista* siempre que la copia *no sea objeto de utilización colectiva ni lucrativa*. No se requiere ser el propietario legítimo de la obra: de ahí que se pague, por ejemplo, el "famoso" canon de los CD y DVD a la SGAE. Si bien, el concepto de *divulgación previa* es relevante. Aunque resulta ambiguo, parece que el legislador hacía referencia a cosas como, por ejemplo, la divulgación de películas por canales públicos de televisión (y su copia en VHS).

Además de estas actividades, se suelen añadir las de penetrar en otros sistemas o dentro de programas, pero éstos suelen tener “nombres propios” (hackers y crackers), como veremos con posterioridad.

7.3.1. Software libre

Ante los precios y lo abusivo de muchas licencias de los programas comerciales existen una serie de programas que se pueden distribuir libremente. Además existen razones filosóficas que se oponen a los sistemas de *copyright* tradicionales:

- No está claro qué es “poseer” una idea o un programa. El sistema tradicional de patentes no parece aplicable aquí.
- Pese a la oposición de las grandes compañías, la distribución libre del software puede aumentar y no disminuir su calidad.
- Los métodos empleados para asegurar las patentes suelen ser “poco ortodoxos”: insultos (“piratería”, “robo”...), exageración de las pérdidas sufridas, redadas, etc.

El software libre es aquel que se distribuye con alguna licencia que permite su *uso*, *estudio* y *modificación* por parte de cualquiera y la *redistribución* a cualquiera. *Linux*, por ejemplo, es un sistema operativo libre. *GNU* es un proyecto para la construcción de herramientas libres.

`http://www.gnu.org/philosophy/free-sw.es.html`

¿Cómo cabe interpretar el concepto “libre”? ¿Software libre es lo mismo que software gratuito? No, es algo más. Algunas empresas cobran por Linux, pero lo hacen a cambio de servicios de valor añadido: sistemas de instalación, paquetes para los programas (que son compatibles entre sí y han sido probados), documentación adicional, consultas durante un período de tiempo, etc. El “modelo de negocio” alrededor del software libre se centra en la prestación de servicios, no en la venta del software. Veamos algunas ventajas del software libre:

- Favorece una “democratización” del software. Mucha gente influye a la hora de decidir las características de los programas.
- Muchas aplicaciones “libres” tienen gran calidad.
- Al estar disponible el código fuente, es fácil corregir errores de programación o recibir asistencia para ello.

Pero también existen algunos problemas:

- Por su origen, suelen ser productos “poco amigables” (pero no es inherente a ellos).

- Suelen requerir más conocimientos por parte del usuario, pero para eso están los "informáticos" ¿no?
- Puede ser difícil justificar su uso en entornos "formales" (empresas), debido por ejemplo a falta de soporte técnico "oficial". Sin embargo, esto está cambiando: *Red Hat* e *IBM* tienen certificaciones para el hardware[6] y dan soporte y asistencia técnica a empresas. Existen, incluso en nuestro país, pequeñas y medianas empresas de servicios informáticos que dan soporte a Linux.

7.3.2. Hacking

Hackers son los que se dedican a conocer íntimamente los programas y los sistemas informáticos y a intentar usarlos de forma innovadora. No tienen, en principio, intención de causar perjuicios. Buscan más una satisfacción personal que una recompensa material. Son individuos que anhelan el conocimiento y bastante "sabios". Va contra su ética alterar cualquier información excepto la de los archivos de registro que puedan contener "rastros" de su paso.

Crackers se dedican a intentar entrar en los sistemas informáticos por la vía de reventar/adivinar claves de acceso de usuarios o bien usar programas con fines "poco claros". Normalmente buscan acceder a información confidencial para su beneficio o bien divertirse destruyendo o alterando la información de un sistema. Muchas veces no está clara la distinción entre un hacker y un cracker, aunque el nivel de conocimiento de los crackers acerca del sistema y su funcionamiento interno suele ser muy bajo. De hecho, pueden aprovechar exploits, agujeros de seguridad. Muchos exploits se publican en la red junto a programas que permiten aprovecharse de ellos. Sin embargo su utilización "infantil" no va a proporcionar ningún conocimiento sobre el sistema ni sobre el programa utilizado en cuestión.

Tipos de licencias

Sin restricciones.
Tipo *shareware*.
Licencia GPL (General Public License) de GNU.
Licencia LGPL (Lesser General Public License) de GNU.
Licencia Artística.
Licencia BSD.
Licencias sin ánimo de lucro.
Licencias de Creative Commons

[6]Certificando que el hardware funciona con el Linux que instalan y además con garantía y asistencia técnica.

Licencias sin restricciones

Normalmente tienen muy pocas o ninguna condición para su redistribución, generalmente basta con citar al autor.

No se paga por el uso.

El código fuente se puede distribuir con el programa, pero no se exige.

Licencias *shareware*

Suele prohibirse el pago por la distribución, aunque muchas veces se incluyen en recopilaciones.

Se suele pedir el pago por el uso comercial o tras un tiempo razonable de prueba (alrededor de 30 días).

Rara vez está disponible el código fuente.

Cuando se paga por la licencia, se suelen obtener ventajas como mejores versiones y documentación.

Licencia GPL de GNU

Permiten la distribución de los programas libremente, pero exigiendo que se distribuyan con el código o se diga cómo obtenerlo. Además debe incluirse una copia de la licencia y no pueden imponerse mayores restricciones que las expresadas por ésta.

Se permite cobrar por los gastos de hacer la copia o servicios adicionales, pero no por el programa en sí.

Se permite modificar el programa, pero las modificaciones que se hagan deben estar también cubiertas por la licencia y ser claramente distinguibles del original. Por tanto, no permite incorporar el programa dentro de programas *no libres* o "propietarios".

El uso del programa no está restringido.

No se da ninguna garantía (aunque los programas suelen funcionar bien).

Existe una versión para documentación y libros.

No se obliga al usuario a aceptar la licencia, salvo en el caso en que quiera modificar o distribuir el software.

Licencia LGPL de GNU

Se utiliza normalmente para *librerías* (colecciones o bibliotecas de rutinas).

El objetivo es "suavizar" o "relajar" las restricciones de la GPL con el objetivo de implantar estándares basados en software libre, favorecer la difusión de partes del mismo y aumentar la competitividad frente a librerías no libres que ofrecen las mismas funcionalidades.

Básicamente permiten usar las librerías (o programas) LGPL en programas no libres.

Hay que ir con cuidado y pensar detenidamente qué programas pueden usar esta licencia, ya que abre una vía para el "aprovechamiento": usar una gran colección de código libre en programas comerciales, por ejemplo.

Licencia Artística

Es una variante de la GPL que permite igualmente el uso, copia y distribución del software junto con el código.

Sin embargo, el propietario del copyright mantiene un control "artístico" (de ahí el nombre) sobre su "obra", esto es, sobre el desarrollo del programa.

Sólo se permiten modificaciones "razonables" (las que el autor entienda como tal).

En cualquier caso se pueden efectuar y distribuir modificaciones al original si: se distribuyen con la versión original, o se incluye el código fuente del programa original claramente diferenciado de las modificaciones, o se usa un nombre distinto para el ejecutable, se documentan las diferencias introducidas y se incluyen las instrucciones necesarias para conseguir el original, o se llega a algún acuerdo con el propietario.

Si es para uso particular, se pueden hacer las modificaciones que se deseen.

Licencia BSD

Al redistribuir el código fuente y/o los ejecutables se debe incluir una copia de la licencia.

No tiene ningún tipo de restricción salvo la de que ni el nombre de la Universidad (California Berkeley), ni el de los autores se pueden usar en la promoción de cualquier producto derivado o que incluya el software.

No obliga a que el software desarrollado a partir de programas con esta licencia esté obligado a incluirla, tenga que ser libre, o BSD, o comercial…

Licencia sin ánimo de lucro

Prohíben expresamente el pago por la distribución.

Suelen incluir el código fuente (en algunos casos es obligatoria la distribución conjunta del código fuente).

El programa se puede utilizar libremente.

Licencias de Creative Commons

Creative Commons es una organización sin ánimo de lucro que pretende ayudar a los autores a publicar su trabajo de manera que queden resguardados sus derechos de autor, y haciendo saber a los demás lo que pueden y lo que no pueden hacer con su obra.

Tienen mucha flexibilidad y abarcan un gran rango de otras licencias (como la GPL y la LGPL de GNU). Proporcionan herramientas y tutoriales que te ayudan a especificar con exactitud los términos de la licencia con los que un autor desea proteger su obra. Para más información, puedes consultar: `http://creativecommons.org/` y `http://creativecommons.org/license/`

PARTE II
MATERIAL COMPLEMENTARIO PARA PRÁCTICAS

Los textos a continuación suponen un manual que contiene el material de apoyo para las prácticas de la asignatura.

Las imágenes carecen de pie de imagen ya que la explicación de la misma se encuentra en los párrafos contiguos, bien anteriores o posteriores, a la misma.

Índice de contenido

CAPÍTULO 1

1. INTRODUCCIÓN A WINDOWS XP

1.1. INTRODUCCIÓN Y OBJETIVOS

Un sistema operativo es un programa que facilita la comunicación entre el usuario y el ordenador con el objeto de que aquél pueda utilizar de la forma más cómoda posible los recursos que el ordenador ofrece. Las principales funciones de un sistema operativo son:

- Servir de intermediario entre los usuarios y la parte material o hardware del ordenador.

- Gestionar y organizar la utilización de los recursos del ordenador (procesador, memoria, discos, periféricos, etc.) entre los diferentes programas que pueden estar ejecutándose.

- Permanecer ejecutándose mientras se utilice el ordenador para poder atender la siguiente tarea que le encomiende el usuario y para responder ante posibles errores o situaciones anómalas.

El sistema operativo que se va a utilizar en esta práctica es WindowsXP, que incorpora una interfaz gráfica orientada a los usuarios. Los sistemas operativos con interfaces gráficas facilitan el uso del ordenador, ya que incorporan un entorno visual que intenta establecer una cierta correspondencia con aspectos relacionados con entornos de trabajo reales y suelen admitir, junto con el teclado, el uso de periféricos apuntadores como el ratón. Gracias a todo ello se evita que el usuario tenga que aprender, recordar y, en determinados casos, teclear el nombre de las operaciones y órdenes que se desean dar al ordenador.

El objetivo de este material es presentar a un nivel meramente introductorio el entorno de trabajo de WindowsXP de forma que el alumno se pueda familiarizar con él. Este cuadernillo puede ser utilizado como apoyo de los ejercicios prácticos.

Hasta ahora Microsoft disponía de dos sistemas operativos diferentes, para el entorno personal o doméstico tenía Windows98 y para el entorno profesional (o de negocios) el WindowsNT/2000. Con WindowsXP se produce una convergencia entre ambas versiones y a partir de ahí se han realizado algunos retoques para diferenciar dos versiones de WindowsXP, una para el ámbito personal llamada WindowsXP Home Edition, y otra para el ámbito profesional denominada WindowsXP Professional. El principal beneficio de esta estrategia para los usuarios domésticos va a ser que WindowsXP ha adquirido la robustez y estabilidad de WindowsNT/2000, esto debe suponer que WindowsXP se quedará menos veces bloqueado, habrá menos ocasiones en la que tengamos que reiniciar el sistema como consecuencia de un error. La mejora para los usuarios profesionales se debe a que WindowsXP tiene mayor compatibilidad con el hardware de la que gozaba WindowsNT/2000.

Las características principales que diferencian este sistema operativo son:

- El sistema de usuarios permite definir varios usuarios con perfiles independientes . Para pasar de un usuario a otro simplemente hay que iniciar

una nueva sesión con otro usuario, más tarde podremos volver a la sesión del primer usuario que permanecerá en el mismo estado que se dejó. El sistema se encarga de manejar a los distintos usuarios activos y sin interferencias.

- El Explorador de Windows está integrado con el Internet Explorer. Ello implica, fundamentalmente, que podemos ver el contenido de las carpetas como si fuese un documento HTML, que podemos navegar (usando la barra de navegación) por el sistema de ficheros y por Internet escribiendo directamente una URL (esto es, la dirección de una página Web) en la caja de texto de las ventanas. Además se pueden grabar CD's con arrastrar y soltar desde el Exporador a la unidad regrabadora, ya no es necesario instalar un programa adicional para grabar CD's. Además, el Explorador de Windows dispone de un menú lateral con las tareas más útiles en función del objeto que tengamos seleccionado en ese momento.

1.1.1. Inicio de Sesión en Windows

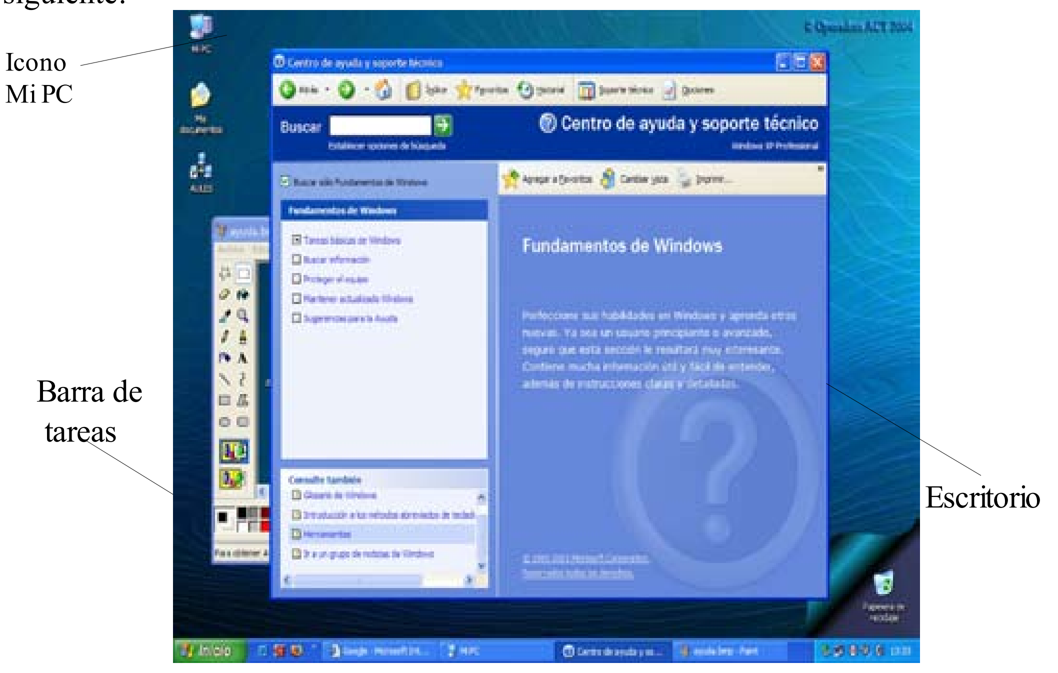
El inicio de sesión de Windows XP del laboratorio de prácticas se realiza de manera autentificada. Esto es, para poder acceder debes introducir tu nombre de usuario y tu contraseña. Si todavía no lo conoces, puedes acceder el usuario genérico que aparece al iniciar la sesión.

Recuerda, que a pesar de entrar con tu usuario, el escritorio se corresponde al PC local y no tu espacio en la máquina ANUBIS (como si ocurre si inicias con LINUX).

1.2. PRELIMINARES

1.2.1. Una primera visión del sistema

La pantalla de un usuario trabajando con el sistema operativo XP es similar a la siguiente:



Los principales elementos que podemos apreciar en la figura son:

■ El escritorio, que es el entorno gráfico que utilizan usuario y ordenador para comunicarse. Ocupa toda la pantalla y sobre él se sitúan otros objetos. Es el nombre genérico que se utiliza para designar la pantalla de fondo sobre la que se disponen los iconos y se ejecutan los programas. El escritorio es el elemento central de WindowsXP.

■ Los iconos, que representan objetos a los que podemos acceder. Por ejemplo: ficheros, archivadores o carpetas, aplicaciones o programas. A cuyas propiedades y contenidos podemos acceder por medio del ratón.

■ La barra de tareas, que aparece por defecto en la parte inferior de la pantalla. Esta se halla dividida en cuatro áreas: botón Inicio, área de lanzadores rápidos, área de programas activos y el área de notificación.

Más adelante estudiaremos con detalle los elementos que acabamos de presentar. Antes vamos a comentar algunos aspectos necesarios para poder desenvolverse en un entorno de trabajo de estas características.

1.2.2. Utilización del ratón

En los sistemas operativos que se apoyan en interfaces gráficas, el dispositivo *ratón* juega siempre un importante papel en la interacción entre usuario y sistema. Utilizando el ratón se pueden llevar a cabo operaciones como seleccionar y mover elementos, abrir o cerrar carpetas, redimensionar ventanas, abrir aplicaciones, etc.

El ratón aparece en la pantalla representado mediante un puntero, normalmente en forma de flecha, que se mueve conforme desplazamos el ratón sobre una superficie. Una vez situado el puntero sobre algún objeto de la pantalla, se pueden pulsar los botones del ratón para indicar la acción que se desea realizar con el objeto seleccionado. Hoy en día existen ratones desde 2 hasta 4 botones, algunos de los cuales a veces se sustituyen por una rueda de scroll. Si bien las funciones básicas son las mismas que la de los ratones con sólo 2 botones (el izquierdo y el derecho) que son las que vamos a explicar aquí.

El botón principal es, por defecto, el izquierdo y permite realizar las acciones más importantes. El botón secundario es el derecho y permite acceder al menú contextual asociado al objeto sobre el que está ubicado el puntero (más adelante se hablará acerca de los menús contextuales). El comportamiento por defecto de los botones del ratón puede cambiarse mediante las opciones de configuración en el Panel de Control).

Las funciones más importantes del ratón son:

■ Apuntar: colocar el puntero del ratón sobre un objeto.

■ Clic: es la acción de pulsar y soltar el botón principal del ratón. Esta acción permite seleccionar objetos.

■ Doble clic: consiste en hacer clic dos veces seguidas. Se debe hacer rápidamente y sin mover el ratón. Mediante doble clic se ejecutan la orden por defecto asociada a los iconos de los objetos (generalmente la primera opción del menú contextual), como por ejemplo abrir carpetas y ejecutar programas.

- Pinchar y Arrastrar: Consiste en apuntar a un objeto, pulsar el botón principal y, sin dejar de pulsar, mover el ratón de forma que el objeto se arrastra por la pantalla. Cuando llegamos a la posición a la que queremos llevar el objeto, soltamos el botón principal, con lo que dejamos el objeto en dicha posición.

Estados del puntero del ratón:

- Normal. El aspecto normal del puntero del ratón es una flecha , cuando el puntero adopta esta forma, indica que el usuario puede realizar cualquier acción (abrir un menú, activar un icono, etc.).

- Ocupado. Tiene la forma de un reloj de arena . El ordenador está procesando y no permite hacer ninguna otra operación hasta que no acabe la actual. El usuario no puede hacer nada con el ratón, excepto moverlo por la pantalla; hay que esperar pacientemente a que el puntero vuelva a su forma Normal

- Texto. Cuando estamos en un lugar en el que es posible escribir texto toma esta forma .

- Punto de inserción. Tiene esta forma . Permite marcar la posición a partir de la cual se van a introducir los caracteres que escribamos con el teclado, es el lugar donde se insertará la próxima letra que escribamos. El punto de inserción se va desplazando según vamos introduciendo texto, también se puede desplazar con las teclas de las flechas del teclado, o bien podemos seleccionar donde insertar texto con un clic con el ratón en la posición deseada.

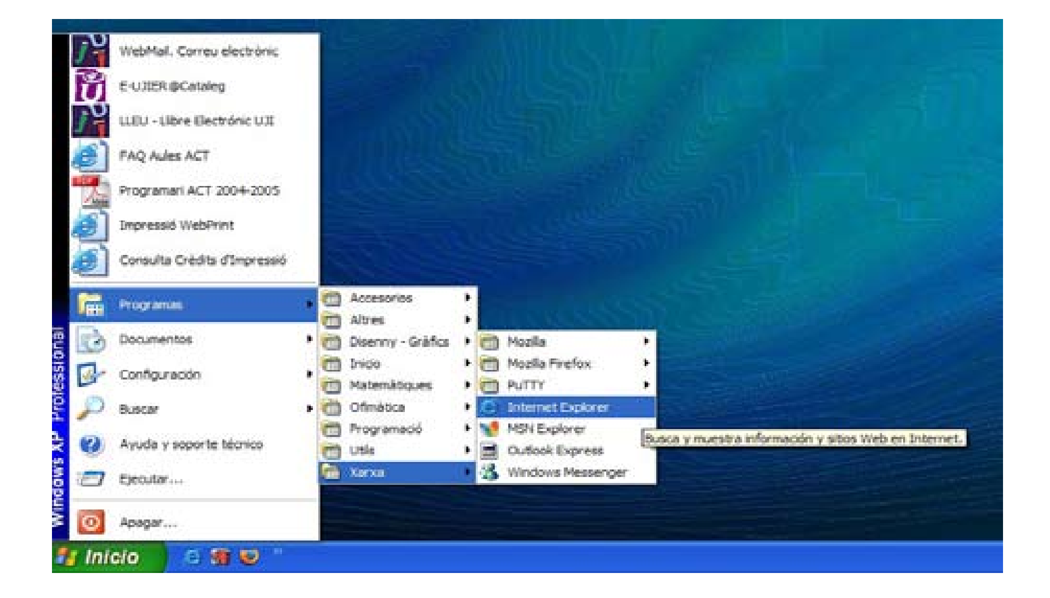
1.2.3. Uso de la Ayuda

La ayuda que incorpora WindowsXP tiene como objetivo intentar resolver algunas de las dudas sobre el manejo de algún componente o herramienta de WindowsXP. Además, las distintas aplicaciones (programas) que se instalan en WindowsXP llevan sus propios temas de ayuda. La ayuda de WindowsXP esta compuesta por una serie de páginas hipertexto por las que te puedes mover de forma similar a como lo haces en una página Web

WindowsXP ofrece tres tipos de ayudas:

- **Tooltips**: al colocar el cursor durante un par de segundos sobre determinados elementos de Windows, como iconos, enlaces, rótulos, etc. Aparece un mensajes con pistas de para que sirve lo seleccionado en forma de pequeñas ventanas de fondo amarillo (figura siguiente).

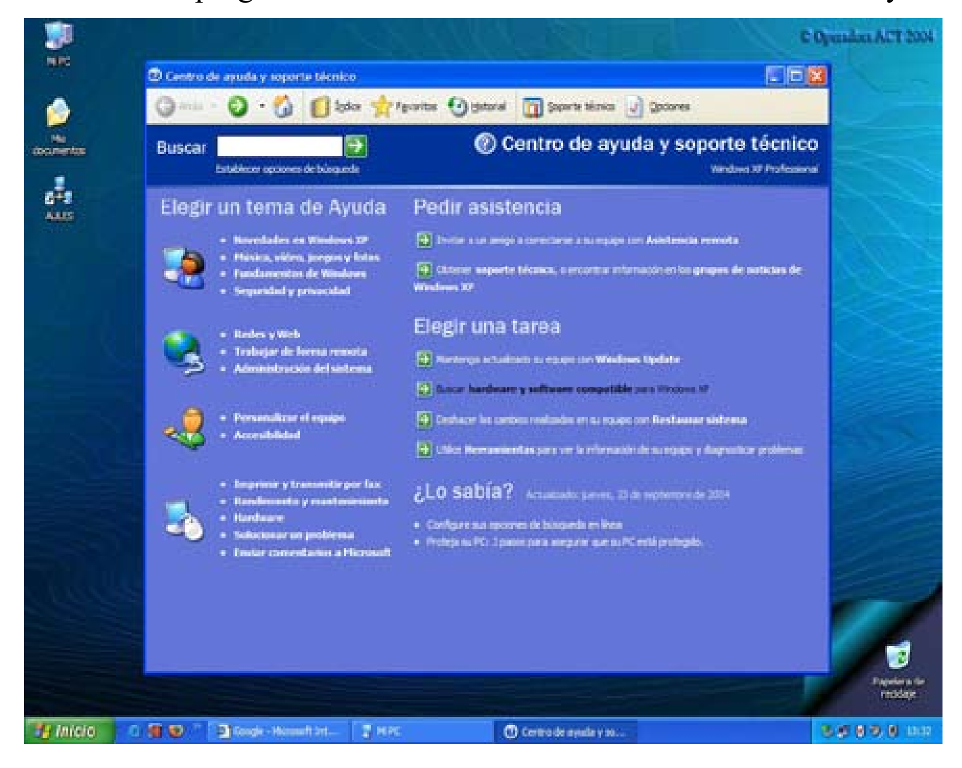
- **¿Qué es esto?**: al pulsar con el botón derecho del ratón en determinados iconos y enlaces, aparece una ventana que dice ¿Qué es esto? al hacer clic en esa ventana se muestra información sobre ese elemento similar al tooltip, pero un poco más amplia.



■ **Centro de ayuda y soporte técnico**: Para conseguir toda la información de ayuda disponible hemos de acceder al Centro de ayuda y soporte técnico (figura siguiente). A este programa se accede:

◆ Con un clic en el botón de Inicio y seleccionando Ayuda y soporte técnico

◆ Desde muchos programas de Windows haciendo clic en el botón de ayuda.



1.2.4. Almacenamiento y gestión de la información

WindowsXP, como la mayoría de los sistemas operativos, organiza los datos que

almacena mediante una estructura arborescente basada en 3 elementos (unidades, carpetas o directorios y ficheros).

1.2.4.1. Las unidades

Un ordenador puede tener conectadas varias unidades (dispositivos) de almacenamiento (disqueteras, disco duro, CD-ROM, etc.) en las que se pueden guardar los datos de forma permanente. Por el contrario, los datos que se almacenan en la memoria principal (RAM, la que utiliza el ordenador para ejecutar los programas) son guardados de forma temporal hasta que finaliza el programa que los maneja o hasta que se apaga el ordenador, momento en el que se pierden. Por esta razón, siempre que se quieran utilizar más adelante los datos que hay en memoria RAM es necesario grabarlos, y no sólo antes de salir de la aplicación que los maneja o de apagar el ordenador, sino también cada cierto tiempo por si se produce un fallo en el sistema o un corte de fluido eléctrico (muchas aplicaciones se pueden configurar para que cada cierto tiempo salven automáticamente el trabajo realizado).

Al haber varias posibles unidades lógicas de almacenamiento permanente (cada una correspondiente a un dispositivo físico), WindowsXP emplea distintas etiquetas para identificar a cada una de ellas, nombrándolas por medio de una letra seguida de dos puntos. Se suele utilizar la siguiente correspondencia entre etiquetas y unidades: las etiquetas A: y B: son para las unidades de disco flexible o disquete (pese a que actualmente los equipos suelen disponer tan sólo de una unidad de este tipo, la A:); la etiqueta C: es la primera unidad de disco duro, etiquetándose los restantes discos duros con las letras siguientes; finalmente, tras los discos duros se emplean las letras posteriores para etiquetar las siguientes unidades, como el lector de CD-ROM.

1.2.4.2. El sistema de ficheros

Al igual que sucede en la mayoría de sistemas operativos, en WindowsXP la información almacenada en los discos que se encuentran en el interior de las unidades se organiza mediante un sistema jerárquico de ficheros y directorios.

Ficheros

Los datos que podemos guardar en las distintas unidades de nuestro ordenador pueden ser muy distintos: los caracteres de una carta escrita con un procesador de textos, los puntos que forman una determinada imagen, las instrucciones que ejecuta un determinado programa, etc. Para poder identificar y utilizar conjuntos de datos que tienen una finalidad única, los agrupamos en ficheros.

En general, podemos dividir los ficheros en 3 tipos: ficheros ejecutables (contienen programas en lenguaje máquina y se pueden ejecutar), ficheros complementarios (contienen información imprescindible para que los ficheros ejecutables puedan funcionar correctamente, por ejemplo, información de configuración del programa) y ficheros de datos (contienen los datos que crea y modifica un determinado programa, como un documento hecho con Word 97, un dibujo hecho con Paint, etc.).

Directorios o carpetas

Para un manejo más sencillo de la información que se almacena en cada disco, los ficheros suelen agruparse siguiendo algún tipo de criterio lógico: ficheros de órdenes del sistema, ficheros de una determinada aplicación, ficheros de un usuario concreto, etc. Pues bien, la forma de agrupar los ficheros en un sistema jerárquico es mediante los directorios. En WindowsXP cada directorio se representa mediante una carpeta que contiene ficheros, o bien contiene otras carpetas (denominadas

subdirectorios), o bien puede contener ambas cosas: ficheros y carpetas. De esta forma los ficheros se pueden organizar de forma eficiente. Si no existiese la posibilidad de crear carpetas, todos los ficheros se tendrían que almacenar juntos en la unidad de almacenamiento correspondiente sin ninguna posibilidad para clasificarlos, lo que sería poco operativo a medida que creciese el número de ficheros. De esta forma, queda completada la estructura arborescente de WindowsXP: existen una o más unidades de disco que contienen una o más carpetas, en las que a su vez se almacenan ficheros y otras carpetas.

Nombres

Tanto las carpetas como los ficheros se identifican mediante un nombre. En WindowsXP, un nombre está formado por una secuencia de hasta 255 caracteres, incluyendo letras, dígitos, caracteres especiales y espacios en blanco. No se permite utilizar en un nombre los siguientes caracteres: barra invertida (\), interrogaciones (¿ y ?), dos puntos (:), asterisco (*), comillas (“), mayor que (>), menor que (<) y barra vertical (|). Los nombres también pueden tener una extensión, separada del nombre por un punto, que puede ser cualquier palabra completa (no se aceptan, en este caso, espacios en blanco). Para mantener compatibilidad con MS-DOS , en WindowsXP se puede utilizar el nombre corto (nombre MS-DOS), que se puede obtener a partir de los nombres normales. Básicamente, el nombre corto, consiste en coger los 6 primeros caracteres del nombre normal, añadiendo detrás el carácter ~ y después un número. Como extensión, se cogen los 3 primeros caracteres de la palabra que hayamos utilizado como extensión (caso que el nombre tenga extensión, si no el nombre corto tampoco tiene extensión).

Rutas de acceso

Por último, un concepto importante más: el de camino o ruta de acceso. Como hemos indicado previamente, un fichero o un directorio se identifican a partir de su nombre y extensión. Sin embargo, en los sistemas jerárquicos es posible tener más de un fichero o directorio con un mismo nombre, siempre que no coincidan en una misma carpeta. Una ruta de acceso es un nombre que designa la posición exacta de un fichero o una carpeta en una unidad de almacenamiento.

La ruta de acceso de un fichero o directorio se construye con la etiqueta de la unidad en la que se encuentra el elemento, seguida de la secuencia de carpetas que permite llegar al elemento en cuestión. Para separar los nombres de las carpetas de la secuencia se utiliza el carácter barra invertida (\). Por ejemplo, la ruta de acceso de un fichero llamado Prac1.doc, situado en la carpeta Documentos en el disquete de 3 y ½ sería: `A:\Documentos\Prac1.doc`. La ruta de acceso de la carpeta Windows, situada en el disco duro, sería: `C:\Windows`.

1.3. ASPECTOS BÁSICOS DE WINDOWS XP

1.3.1. Las ventanas

A continuación vamos a destacar los distintos elementos que nos podemos encontrar en las ventanas y sus funcionalidades.

- La **barra de título** (barra superior) contiene el nombre de la ventana. Arrastrándola podemos desplazar la ventana sobre el escritorio.

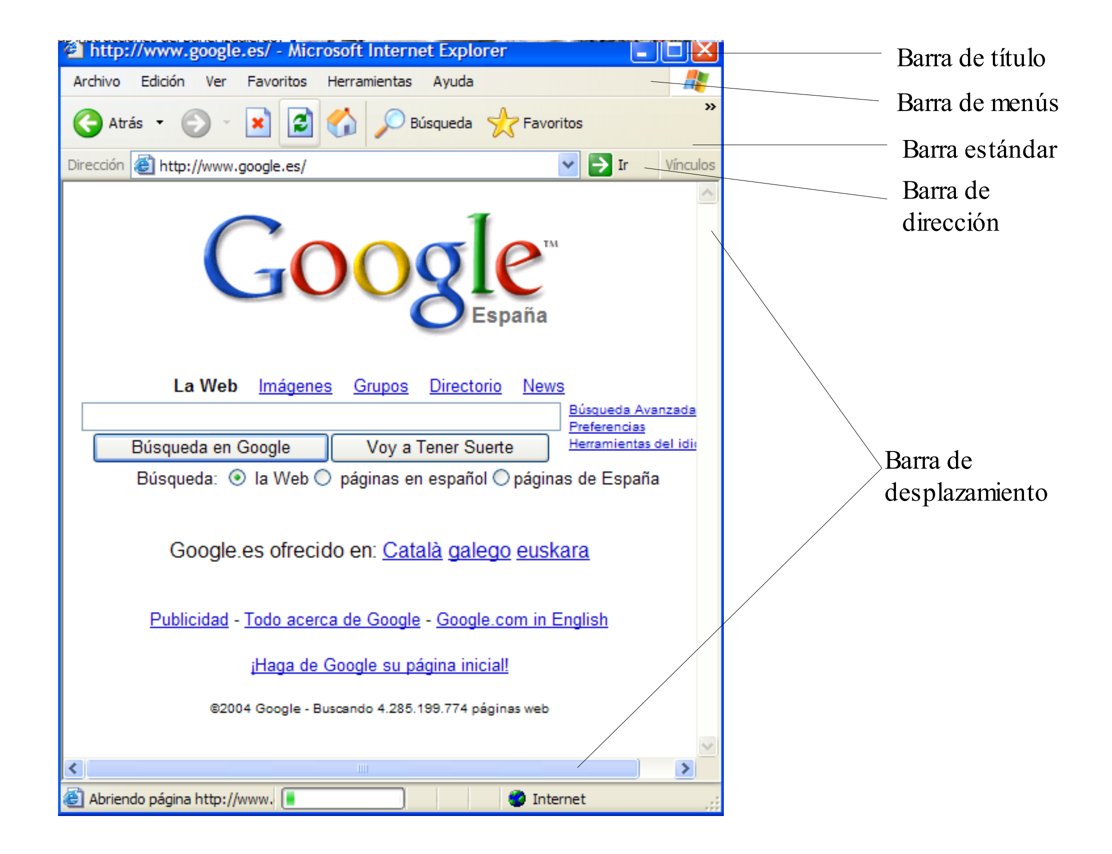
- La **barra de menús** permite acceder a una serie de órdenes con las que

realizar acciones relacionadas con la ventana y su contenido. Al pinchar en cada menú se abrirán las opciones que lo componen

■ La **barra estándar** contiene botones para las operaciones más utilizadas de la barra de menú, de esta forma nos agiliza las operaciones.

■ La **barra de direcciones**. Es una caja de texto, donde se puede escribir la ruta de un fichero o una URL (dirección de Internet).

■ Las **barras de desplazamiento** (barras lateral e inferior con flechas) permiten ver el contenido de la ventana que no cabe en la pantalla. Si no está visible esta barra selecciona del menú Ver, la opción Barra de Herramientas, a continuación selecciona la opción Estándar



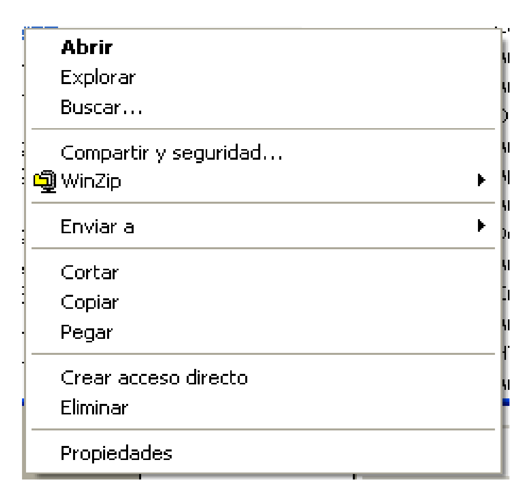
■ La **barra de estado** (barra inferior) muestra información adicional. Esta barra es opcional, para activarla ir al menú Ver, y seleccionar en barra de estado. Si tenemos seleccionada una carpeta, en la barra se muestra el número de objetos que contiene la carpeta seleccionada y el espacio de almacenamiento (en disco) que éstos ocupan. Esta información también aparece en la parte izquierda de la ventana si cabe con más detalle (si tenemos activada la opción de visualizar el contenido de las ventanas como HTML, opción por defecto en la instalación de WindowsXP. En la próxima figura tenemos abierto el Internet Explorer, y podemos apreciar que en la barra de estado se nos da información del estado en que se encuentra la pagina Web solicitada.

■ Arrastrando los **bordes de la ventana** podemos cambiar sus dimensiones. Arrastrando sobre las esquinas de la ventana es posible modificar dos dimensiones simultáneamente.

■ Cuadros de ***minimizar, maximizar*** y ***cerrar*** ventana (esquina derecha de la barra de título). En la barra de título aparecen estos tres cuadraditos que permiten: en el caso de minimizar, que la ventana desaparezca del escritorio pero que siga activa (la seguimos viendo en la barra de tareas y si pulsamos su icono en esta barra, la ventana vuelve a aparecer); el botón de maximizar cambia la dimensión de la ventana, generalmente para que ocupe toda la pantalla; el botón de cerrar cierra la ventana.

1.3.2. Los menús contextuales

Cualquier objeto que haya en el escritorio tiene asociado un menú contextual similar al de la figura inferior que se activa con un clic del botón secundario del ratón sobre el objeto. Para ver el menú contextual del escritorio o de la barra de tareas basta con hacer esta operación en cualquier parte vacía del mismo.



Todos los objetos de WindowsXP tienen su propio menú contextual, y las opciones que aparecen dependen del tipo de objeto. Serán distintas si el objeto es un documento Word, si es un programa, si es una carpeta, etc. Además, es interesante remarcar que, a través de la opción propiedades del menú contextual, se pueden conocer datos como la capacidad y ocupación de las unidades de almacenamiento secundario, o la dimensión y fechas de creación o modificación de carpetas o ficheros o los permisos del fichero o carpeta.

1.3.3. Los accesos directos

Los accesos directos facilitan el acceso a las carpetas, los programas y los documentos que se usan con más frecuencia. Un acceso directo a un objeto es un icono que se puede colocar en el escritorio o en cualquier carpeta y que permite ejecutarlo (en el caso en que el objeto original sea un programa) o abrirlo (si el objeto fuera una carpeta) sin necesidad de buscar el objeto original en el sistema de directorios. Un acceso directo no cambia la ubicación del objeto, y su eliminación no implica la eliminación del original. Se puede crear un acceso directo para cualquier elemento, incluyendo carpetas, unidades de disco, otros equipos e impresoras.

Se puede crear un acceso directo:

■ a partir del menú contextual del objeto o,

■ arrastrando el objeto donde queremos situar el acceso directo o,

■ directamente con el menú contextual del escritorio usando la opción **Nuevo Acceso directo**

En el menú contextual del acceso directo, podemos ver sus propiedades, en las que cabe destacar la propiedad de **destino** donde se indica la dirección del elemento al cual se accede con ese acceso directo.

1.3.4. Carpetas especiales

Al abrir una carpeta, observaremos que en la ventana aparece un menú lateral con las tareas más útiles en función del objeto que tengamos seleccionado en ese momento, como se observa en la figura posterior.

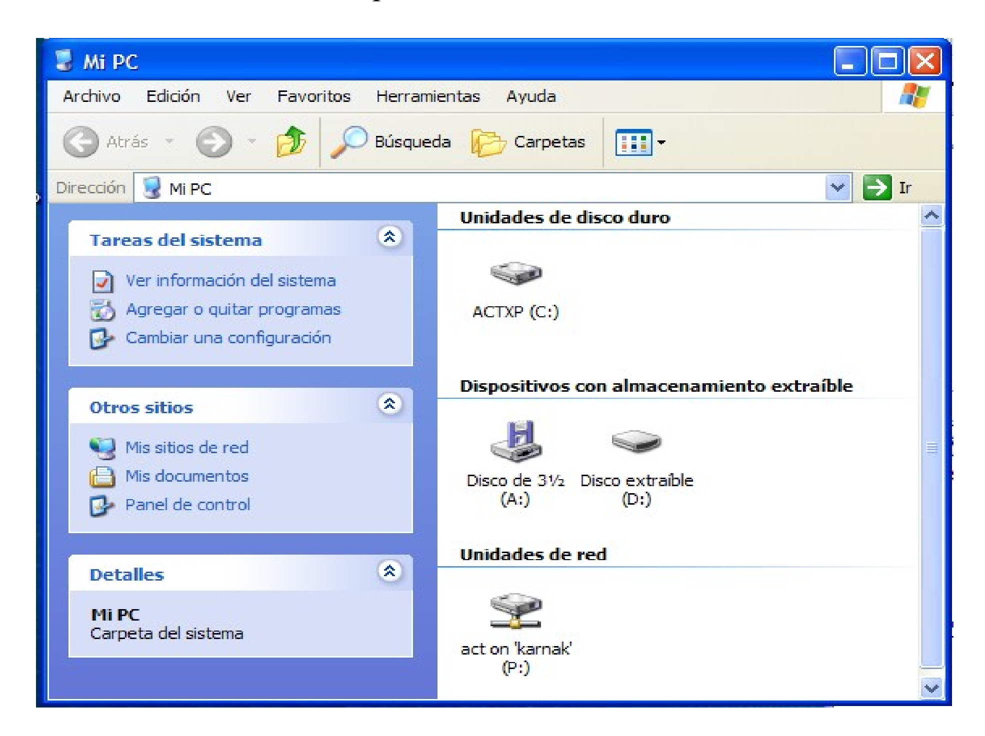
1.3.4.1. La papelera de reciclaje

Para borrar cualquier icono (carpeta o fichero) del ordenador, basta con arrastrarlo con el ratón y soltarlo en la papelera de reciclaje. Realmente, al arrastrar un elemento a la Papelera no se borra físicamente dicho elemento. Para ello, hay que acceder al menú contextual de la papelera de reciclaje y seleccionar vaciar papelera de reciclaje. El borrado de un acceso directo a un elemento destino, esto no implica el borrado del elemento de destino, sólo borra su acceso directo.

1.3.4.2. Mi PC

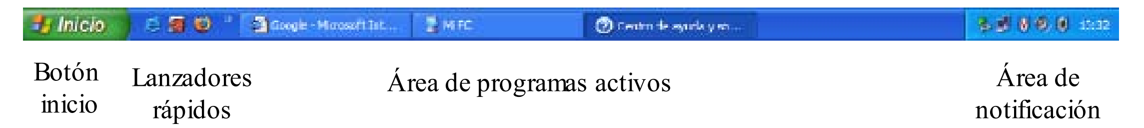
La carpeta de Mi PC es la carpeta principal, en la cual se encuentran todas las unidades a las cuales tenemos acceso en nuestro ordenador: discos duros, en este caso solo disponemos de un disco duro etiquetado como `ACTXP(C:)`, discos extraíbles, como la disquetera (`a:`) y el lápiz de memoria (`D:`) y las unidad de red (ordenadores a los cuales estamos conectados) `karnak(P:)`.

En el menú lateral aparecen ciertas tareas que el sistema decide que son útiles en función de la carpeta que tenemos abierta. En este caso nos aparecen tres grupos de opciones en el menú, la realización de tareas del sistema, el acceso a otras carpetas y finalmente detalles sobre la carpeta.



1.3.5. La barra de tareas

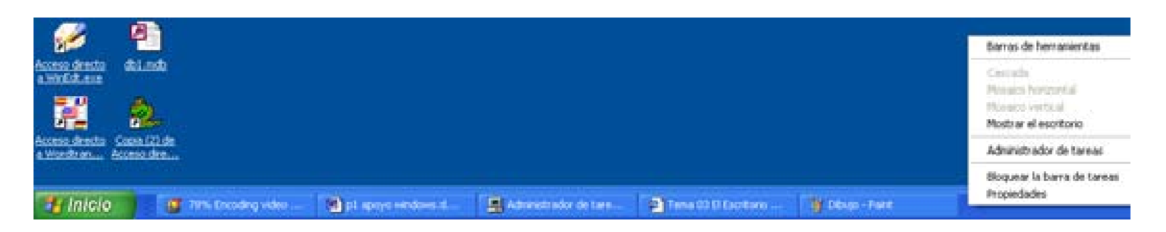
La barra de tareas es un elemento muy importante. Normalmente se presenta como una franja horizontal en la parte inferior del escritorio y con el siguiente aspecto:



Como vemos en la figura anterior, la *barra de tareas* esta dividida en 4 zonas: *botón Inicio*, *área de lanzadores rápidos*, área de programas activos y el área de notificación. El *botón Inicio* es una aplicación que se puede configurar y utilizar como un método rápido para iniciar un programa o encontrar un fichero y para salir de la sesión o apagar el equipo. El *área de lanzadores rápidos* permite colocar en la barra de tareas accesos directos a los programas que seleccione el usuario, para que sean más accesibles. El *área de programas activos* contiene los iconos de las aplicaciones que tenemos ejecutándose y permite cambiar de aplicación activa o cerrar la aplicación. El *área de notificación* ofrece información acerca de determinadas características del sistema, incluida la hora del ordenador y las tareas pendientes en las impresoras.

Cada vez que se abre una ventana (ya sea al ejecutar un programa o al abrir una carpeta) se añade a la barra de tareas un icono (rectangular) que representa dicha ventana. Al hacer clic con el ratón en un determinado icono, la ventana o el programa asociados a él se activan, es decir, pasan a ser los elementos sobre los que puede trabajar el usuario. De esta forma, la barra de tareas permite conmutar cómodamente entre todas las ventanas y aplicaciones abiertas en WindowsXP. Si la ventana está abierta en el escritorio, pero en segundo plano (detrás de otras ventanas), al hacer clic sobre su icono en la barra de tareas se convierte en la ventana activa. Si la ventana está abierta pero minimizada, al hacer clic sobre el icono se restaura y queda como ventana activa. Si se pulsa con el botón secundario sobre el icono aparece el menú contextual asociado a la ventana.

Por su parte, desde el menú contextual de la barra de tareas se puede acceder a algunas opciones interesantes de organización de las aplicaciones y ventanas activas, como son: **Cascada**, **Mosaico horizontal**, **Mosaico vertical** y **Minimizar todas las ventanas**:



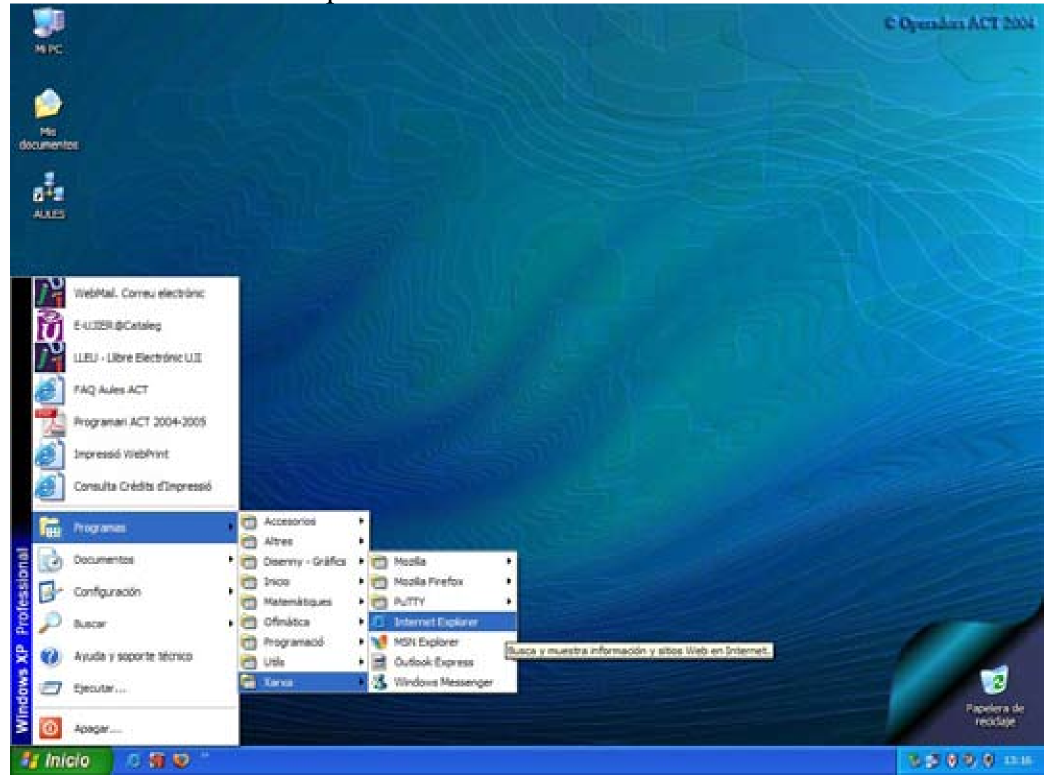
Las tres primeras opciones organizan automáticamente las ventanas abiertas que no están minimizadas. El modo cascada superpone una ventana encima de otra; el mosaico horizontal divide toda la pantalla en franjas horizontales, asignándole una franja a cada ventana abierta; y el mosaico vertical es lo mismo que el horizontal, pero dividiendo la pantalla en franjas verticales. La opción **Minimizar todas las**

ventanas minimiza todas las ventanas abiertas en el escritorio. Cuando se ha activado esta opción, en el menú contextual de la barra de tareas aparecerá una nueva opción, **Deshacer minimizar todo**, que restaura todas las ventanas a su posición y forma anterior. Esta opción es muy útil para ocultar rápidamente todas las ventanas abiertas y poder encontrar un icono del escritorio.

1.3.6. El botón Inicio

La función principal de este botón es permitir el acceso cómodo a los programas y utilidades que hay instalados en el ordenador. Al hacer clic en este botón se despliega el menú que tiene asociado, cuyo contenido puede ser similar al de la figura de la próxima página.

La opción de **Ayuda** la hemos estudiado anteriormente, la de **Configuración** permite hacer cambios en el entorno del sistema, y la de **Apagar el sistema** que permite apagar correctamente el equipo, la veremos más adelante. Vamos a centrarnos en las restantes posibilidades.

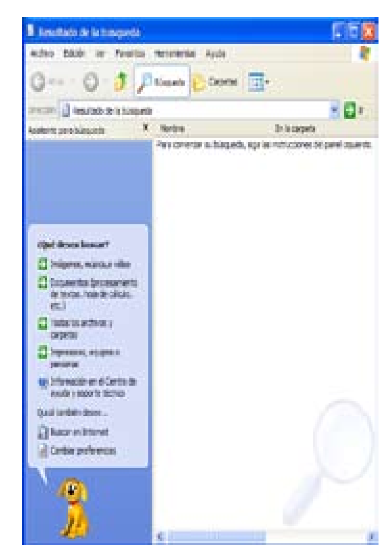


1.3.6.1. Programas

La opción Programas permite acceder y ejecutar la mayoría de programas instalados en WindowsXP (aquellos que han creado la correspondiente "entrada" en este menú al instalarse). A través de esta opción se despliega una estructura de menús y submenús desde los cuales podemos ejecutar varias aplicaciones. Veáse como ejemplo la figura anterior extraída de nuestro ordenador.

1.3.6.2. Documentos

La opción **Documentos** contiene una lista de los últimos 15 documentos (ficheros de datos) que se han abierto (independientemente de la aplicación utilizada para editarlos). En la lista aparece el nombre del documento y un icono que identifica el tipo del documento. Al hacer clic sobre uno de ellos se ejecuta automáticamente el programa asociado (el que se utilizó para crearlo) y se abre el documento. Sólo se añaden a esta lista aquellos documentos que se han abierto, bien directamente desde el escritorio, bien utilizando la opción **Abrir** del menú **Archivo** de una aplicación diseñada específicamente para WindowsXP. Cuando la lista ya posee 15 elementos, cada vez que se abra un documento, se borra el más antiguo de esta lista, sustituyéndose por el nuevo.

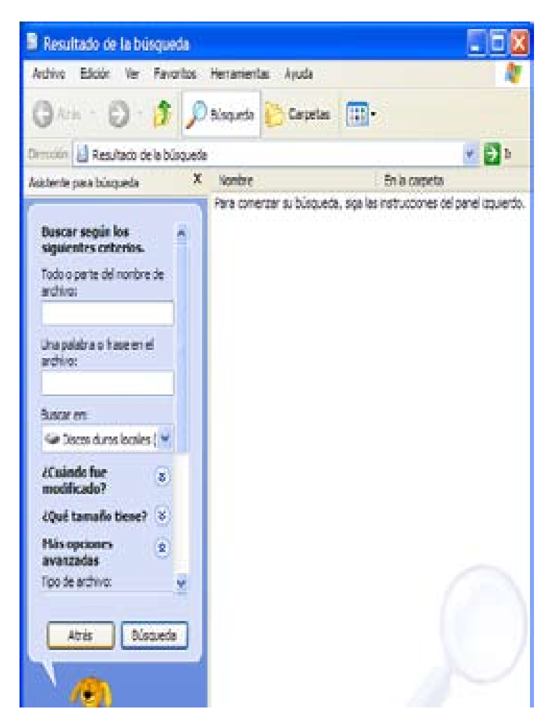


1.3.6.3. Buscar

La función de Búsqueda se incorporó en Windows para permitir encontrar ficheros especificando ciertos parámetros de búsqueda. WindowsXP también permite buscar equipos en la red, personas en la libreta de direcciones, páginas en Internet. Para acceder a la función búsqueda podemos:

- pulsar en Inicio y seleccionar Buscar,

- desde la barra de estándar de las ventanas de Windows, seleccionando el icono de Búsqueda.



En ambos casos se desplegará la ventana general de búsqueda que mostramos en la figura de la derecha. En primer lugar debemos elegir una de las opciones que se nos presentan según lo que queramos realizar. Las tres primeras son similares y permiten buscar ficheros o información en el equipo

A modo de ejemplo vamos a ver como funciona la opción **Buscar Todos los Documentos** (al seleccionar esta opción del menú lateral, nos aparece una nueva ventana como la de la figura de la izquierda. Ahora en el menú lateral se nos solicitan tres campos de búsqueda básicos: nombre, palabra existente en el documento y la ubicación (ruta) donde buscar el documento. No es necesario rellenar todos los campos.

Además de los campos de búsqueda

básicos, el menú lateral nos permite acceder a otros campos para dar más información sobre el documento que deseamos buscar.

Para que el sistema comience la búsqueda, tras introducir los parámetros deseados deberemos hacer clic en el botón **búsqueda** de modo que en el área de la derecha de la ventana, aparecerá un listado con los ficheros que cumplen los requisitos.

En el campo Nombre, se puede escribir el nombre completo del documento a buscar o un *patrón de texto*. Los patrones son cadenas de caracteres que pueden incluir caracteres comodín para designar todos los nombres que tienen una determinada característica.

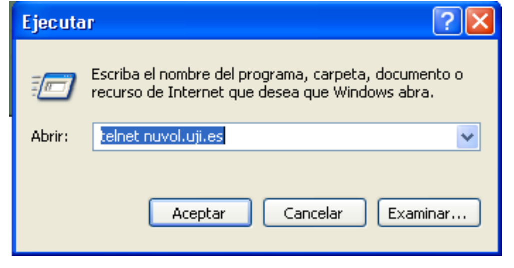
Existen dos caracteres comodines:

- ***** : representa cualquier cadena de caracteres de cualquier longitud (incluso cero). Por ejemplo, a* es un patrón que se ajusta a cualquier palabra que empieza por a. La palabra a también se incluiría ya que la cadena de longitud 0 es la cadena vacía.

- **?** : representa cualquier carácter (pero siempre uno). Por ejemplo, algunas palabras que se ajustan al patrón C?lc.exe son Calc.exe, Cilc.exe, Cxlc.exe, Crlc.exe, etc., es decir, aquellas palabras que en lugar del carácter comodín tienen un solo carácter, sea el que sea.

1.3.6.4. Ejecutar

La opción **Ejecutar** permite ejecutar programas de forma directa, sin tener que recurrir a hacer doble clic sobre su icono o a tener que buscarlo a partir del submenú **Programas** del **botón Inicio**. También es una alternativa a crear accesos directos a programas en el escritorio, aunque es más incómodo ya que hay que conocer la posición exacta (la ruta de acceso) del programa que se quiere ejecutar. Al hacer clic en esta opción aparecerá una ventana como la de la figura de la izquierda:



Por defecto, aparece resaltado el programa que se ejecutó la última vez que se utilizó esta opción. Al pulsar **Examinar…** es posible buscar el programa que se desea ejecutar.

1.3.7. El explorador de Windows

El explorador de Windows básicamente está orientado a facilitar la realización de tareas relacionadas con la organización de la información que se encuentra almacenada en los sistemas de ficheros y directorios de las distintas unidades: copiar, trasladar y eliminar ficheros y carpetas entre dispositivos o en una misma unidad, crear directorios y accesos directos, acceder a menús contextuales de los objetos, ejecutar programas, etc.

En realidad, todo lo que se puede hacer con el explorador de archivos, se puede hacer manipulando carpetas y ficheros tomando como punto de partida la carpeta Mi PC o Mis documentos. La ventaja del explorador es que presenta la información de una forma más cómoda y compacta, mediante dos paneles dentro de la ventana. El

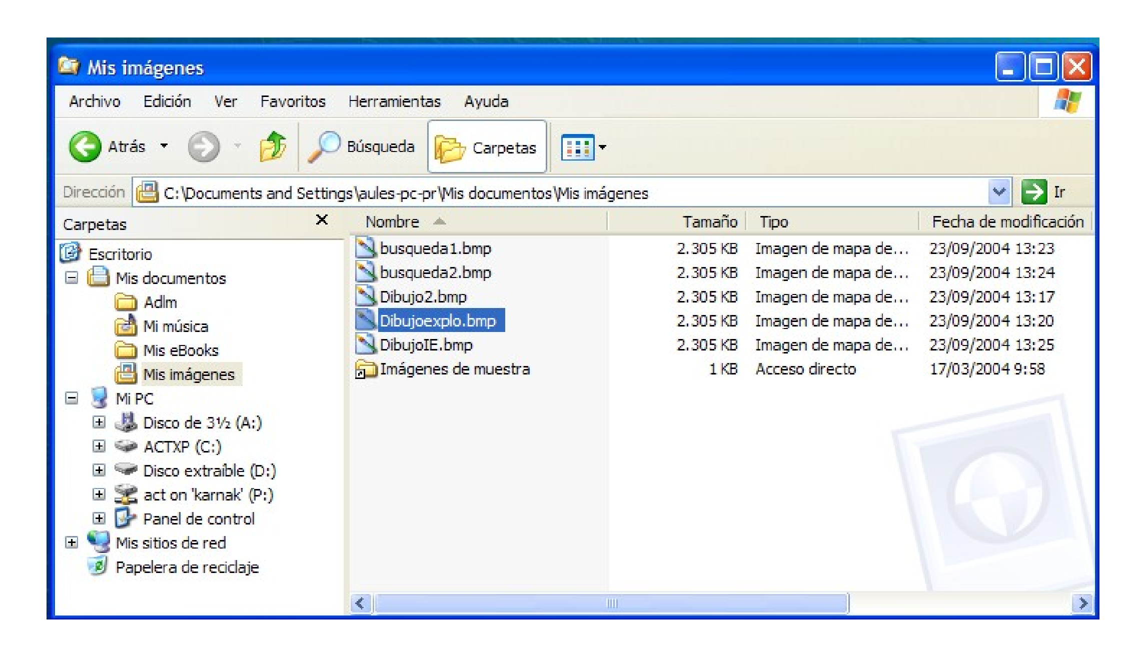
panel de la izquierda muestra el árbol (la jerarquía) de unidades, carpetas y subcarpetas (no aparecen archivos ni accesos directos). El panel de la derecha muestra el contenido de la unidad o carpeta seleccionada en el otro panel, incluyendo sus ficheros, carpetas o accesos directos, tal y como se muestra en la siguiente figura.

Sobre el panel izquierdo podemos realizar las siguientes acciones:

- Clic sobre un símbolo + del panel izquierdo: se abren la unidad o carpeta implicada apareciendo en la jerarquía de directorios del panel izquierdo las subcarpetas que contiene.

- Clic sobre el símbolo – : se cierra la carpeta y desaparecen sus subcarpetas.

- Seleccionar el nombre de una unidad o carpeta: aparece en el panel derecho el contenido de ese elemento.



Para iniciar el Explorador de Windows

- Seleccionar en la barra de tareas o del escritorio el icono del explorador

- Pulsar inicio y Seleccionar Todos los programas, selecciona Accesorios, y para finalizar selecciona el Explorador de Windows

Si en vez de abrir directamente el explorador, abrimos una carpeta en una ventana, por ejemplo la carpeta de : Mis documentos se accede al explorador de Windows pero se muestra una ventana similar a la que hemos visto con la carpeta de Mi PC, con un menú lateral en la zona de la izquierda en vez del árbol de directorios. Y en la zona de la derecha, nos aparecen directamente los ficheros, carpetas y accesos directos que contiene la carpeta que hemos seleccionado.

Vamos a ver con un poco más de detalle que hacen los botones de la **barra estándar**:

- El botón atrás Atrás nos permitirá ir a la última página que hayamos visto.

- El botón adelante uando esta activo, permite ir una página hacia adelante.

- El botón arriba nos permitirá subir de nivel, es decir, situarnos en la carpeta que contiene la carpeta actual.

- El botón de búsqueda Búsqueda nos permite acceder directamente a la opción de búsqueda de WindowsXP

- El botón carpetas Carpetas permite que en la zona de la ventana aparezca bien el árbol de directorios o bien el menú lateral las tareas más frecuentes según el archivo o carpeta que tengamos seleccionado.

- El botón vistas nos permite cambiar las vistas de las carpetas (vista detalle, vista iconos grandes,...). Según el tipo de vista que tengamos activado veremos distinto tipo de información sobre los archivos, si tenemos activada la vista Detalles (como en la figura anterior) veremos: el nombre, tamaño, tipo y fecha de modificación de cada archivo o carpeta

Con el menú **Archivo**, seleccionado un objeto del panel de la derecha podremos: **Crear un acceso directo**, **Eliminar** (enviar a la Papelera de reciclaje), **Cambiar nombre** o ver las **Propiedades** de un objeto previamente seleccionado, (en todos estos casos también podemos hacer uso del menú contextual).

Si se desea seleccionar varios elementos cuyos iconos aparecen de forma consecutiva en el panel derecho del explorador, se debe hacer clic en el icono del primer elemento del grupo que se quiere seleccionar y, mientras se mantiene pulsada la tecla de Mayúsculas, hacer clic en el icono del último elemento del grupo que se quiere seleccionar. Si lo que se pretende es seleccionar varios elementos que no aparecen de forma consecutiva, entonces se debe hacer clic en un primer elemento (cualquiera de los que se quieran seleccionar) y extender la selección (ir incorporando nuevos elementos a la selección) haciendo clic sobre los restantes elementos al mismo tiempo que se pulsa la tecla *CONTROL*

Con el menú **Edición** podremos efectuar cualquier operación con ficheros y carpetas seleccionados: copiar y/o mover..Una vez seleccionados el o los elementos con los que se desea trabajar (copiarlos o trasladarlos –moverlos–) hay que ejecutar las opciones **Copiar** (para copiar) o **Cortar** (para mover) y seleccionar la carpeta donde se quieren copiar o mover los elementos seleccionados y ejecutar la opción **Pegar** del menú **Edición**.

En WindowsXP se puede seleccionar un grupo de elementos de una forma más sencilla. Basta con arrastrar el ratón sobre el área donde se encuentran los objetos a seleccionar. Al soltar, todos los objetos cuyos iconos están dentro del área marcada con el ratón aparecen seleccionados.

También se puede aplicar esta misma estrategia para copiar y trasladar objetos. Basta con arrastrar los iconos previamente seleccionados al icono de la carpeta o dispositivo donde los queramos depositar. Si la carpeta destino está en la misma unidad de disco, los objetos se trasladan; si está en otra unidad, se copian. Para copiar elementos en diferentes directorios de la misma unidad, hay que trasladarlos

con el ratón a la vez que se pulsa la tecla de *CONTROL*.

1.4. 7-ZIP: COMPRESIÓN Y DESCOMPRESIÓN DE FICHEROS

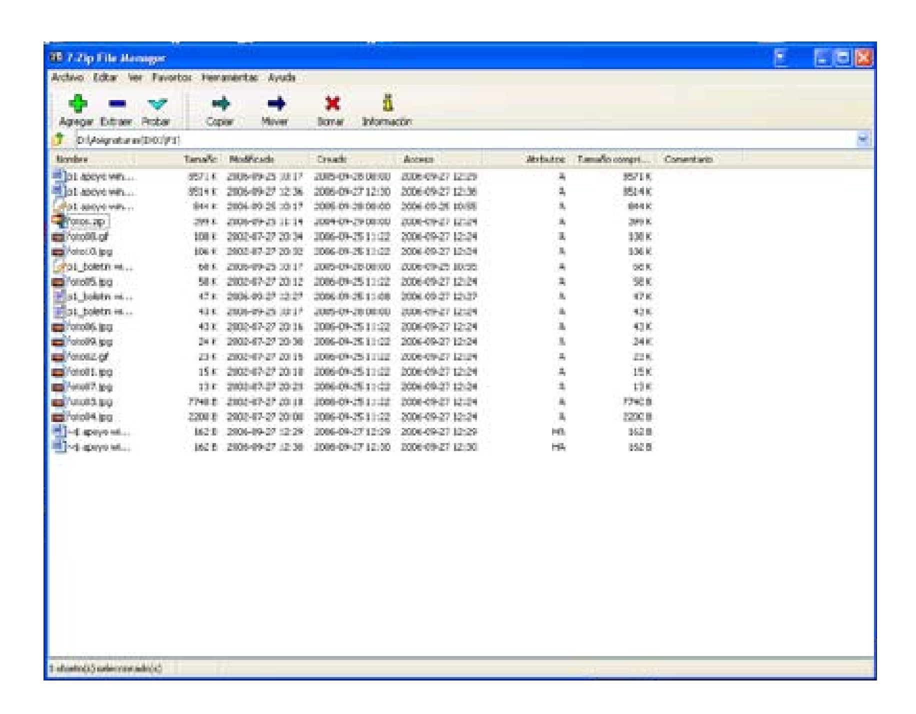
7-ZIP es una herramienta que permite trabajar cómodamente con archivos comprimidos. Es un programa ajeno a Windows, desarrollado bajo licencia *GPL*, y tiene la gran ventaja de que el formato de compresión que utiliza es compatible con otras herramientas software gratuitas para diferentes sistemas (por ejemplo gzip en Linux).

Cuando se está hablando de compresión de ficheros, es necesario ser estricto con algunos términos ya que introducen matices en su significado que no aparecen habitualmente al utilizarlos en otros contextos. Por ejemplo, la palabra archivo se refiere a un "fichero que contiene otros ficheros", y generalmente estos ficheros incluidos en un archivo están comprimidos. Particularmente en *7-ZIP,* el término agregar se emplea con el significado de "comprimir ficheros y añadirlos a un archivo", y extraer significa "descomprimir ficheros que están en un archivo y crear ficheros separados en el disco".

Generalmente, la herramienta *7-ZIP* se puede ejecutar desde el menú **Inicio**. La opción que inicia su ejecución puede aparecer directamente en el menú **Inicio**, o en el menú **Inicio→Programas**, o incluso en algún otro submenú de **Inicio→ Programas**. Si no se encuentra aquí, se puede localizar mediante la herramienta **Buscar** del menú **Inicio**, buscando 7-zip*.exe ya que el nombre del programa puede estar acompañado de un número

1.4.1. La ventana de 7-ZIP

Al ejecutar el programa 7-Zip nos aparecerá una ventana parecida a la que se muestra a continuación:



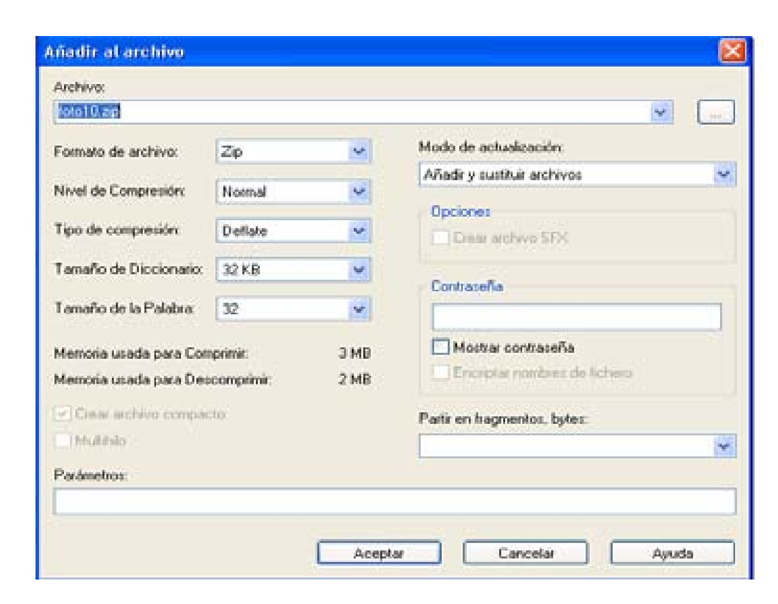
La ventana de 7-ZIP incluye los componentes habituales de las ventanas de Windows. En particular, obsérvese la barra de herramientas debajo de los menús, la cual proporciona formas rápidas de realizar las operaciones más frecuentes, el área principal (en blanco) en la zona central, que se utiliza para mostrar información individual sobre los ficheros contenidos en un archivo, y la barra de estado en la parte inferior, la cual se utiliza para mostrar información general del archivo.

1.4.2. Adición de ficheros a un archivo

La adición de ficheros se hace desde la pantalla inicial y un cuadro de dialogo que se mostrará al hacer clic en el botón **Agregar** de la barra de herramientas o al seleccionar **Añadir al archivo** en el menú desplegable **Archivo→7-ZIP**.

En la pantalla inicial se deben seleccionar los ficheros que se quiere añadir al archivo. Para seleccionar un único fichero se hará clic sobre su icono. Para seleccionar más de un fichero se pueden utilizar métodos convencionales de Windows, como hacer clic sobre un primer fichero y hacer clic sobre un último fichero mientras se tiene pulsada la tecla *MAYÚSCULAS* para seleccionar todos los ficheros de la lista que están entre el primero y el último, o como mantener pulsada la tecla *CONTROL* mientras se van haciendo clics en ficheros salteados de la lista para seleccionar exclusivamente aquellos que en los que se ha hecho clic. Además, se puede ir cambiando de directorios para seleccionar ficheros en distintos directorios. Existen más opciones que se pueden elegir en este cuadro de diálogo pero no las estudiaremos.

Una vez que se ha hecho la selección de todos los ficheros, se debe hacer clic en la opción **Agregar** para añadirlos al archivo. En ese momento aparece el cuadro de dialogo que se muestra a continuación, y en el que hay que seleccionar el archivo donde queremos comprimir los archivos y el tipo de compresión.



Si se desea comprimir un fichero o directorio grande y guardarlo en discos flexibles (disquetes), bastará con especificarle a 7-ZIP, en el momento de agregar los nuevos archivos, el tamaño deseado en el cuadro de texto Partir en fragmentos, bytes.

1.4.3. Extracción de ficheros de un archivo

Cuando se extrae un fichero, 7-ZIP lo descomprime y lo sitúa donde se seleccione. Se pueden restaurar directorios completos y mantener su estructura de subdirectorios.

Para extraer desde un archivo se puede hacer clic en el botón Extraer de la barra de herramientas o seleccionar Extraer... en el menú desplegable Archivo. De ambas maneras se activará el mismo cuadro de diálogo. En él se puede seleccionar el directorio en el que se situarán los ficheros que se extraigan. También se puede indicar si se extraerán todos los archivos o sólo algunos.

1.4.4. Eliminación de ficheros de un archivo

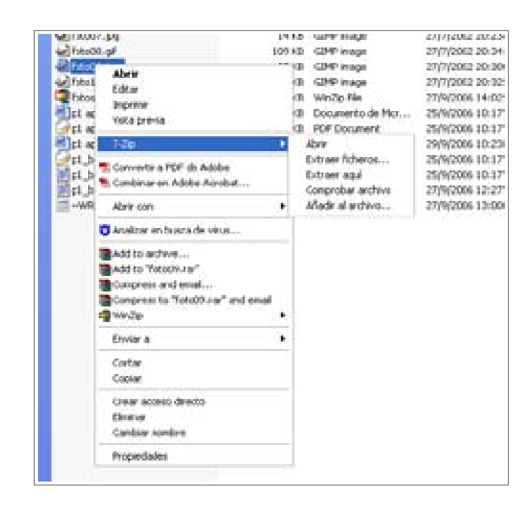
7-ZIP puede eliminar ficheros seleccionados de un archivo. Para ello se debe seleccionar los ficheros a eliminar y pulsar el botón Borrar de la barra de herramientas. Se activará un cuadro de diálogo en el que se podrá confirmar si se desea eliminar el fichero.

1.4.5. Formas rápidas de compresión y descompresión

También se puede comprimir y descomprimir ficheros y directorios con 7-ZIP desde sus menús contextuales. En ellos aparecen opciones para comprimir si el fichero o directorio no está comprimido, y para descomprimir cuando el menú contextual corresponde a un archivo con ficheros comprimidos.

Si se activa el menú contextual de un fichero o directorio (en el escritorio, dentro de una ventana o en el explorador de Windows) aparecerá un menú como el de la izquierda. En él aparecen varias opciones:

- Abrir: Permite abrir un fichero zip.
- Extraer ficheros: Abre el dialogo extraer lo que permite extraer el fichero seleccionado.
- Extraer Aquí: extrae el fichero seleccionado en el directorio actual.
- Comprobar archivo: verifica que el archivo comprimido esta correcto.
- Añadir Archivo: Abre el dialogo agregar lo que permite agregar el fichero seleccionado a un fichero zip



1.5. TERMINAR UNA SESIÓN DE TRABAJO

Cuando un usuario finaliza su trabajo debe salir del ordenador haciendo uso de las opciones que le da la orden asociada al botón Inicio- Apagar el equipo:

- Suspender permite dejar "dormido" el ordenador, de forma que aparentemente simula estar apagado, aunque mantiene activas todas las aplicaciones que estuvieran en marcha. Esta opción es útil cuando el usuario que está utilizando el ordenador va a dejar de emplearlo durante un rato, ya que permite ahorrar energía y a la vez facilita una rápida reanudación del trabajo (sin tener que esperar al, muchas veces interminable, proceso de arranque de Windows). Para reactivar el ordenador normalmente bastará con mover el ratón o pulsar una tecla.

- Apagar el sistema significa apagar el ordenador. Esta operación no es inmediata ya que el sistema operativo realiza una serie de tareas antes de permitir desconectarlo. Cuando seleccionamos esta opción pasa un cierto tiempo hasta que el propio WindowsXP nos indica que lo paremos físicamente. En algunos ordenadores (como es el caso de los PCs de prácticas) se puede configurar el sistema para que se apague automáticamente. Solamente hemos de preocuparnos de apagar el monitor.

- Reiniciar significa que el ordenador parará y automáticamente volverá a arrancar.

1.5.1. Conclusión: un ejercicio de repaso

Abre la carpeta Mi PC del escritorio. Después abre la unidad `C:` y, dentro de esta unidad, la carpeta Windows. Observa el efecto en la barra de tareas. Cambia las dimensiones de las ventanas, desplázalas por el escritorio, minimízalas y maximízalas. Ciérralas todas.

Utiliza la opción **Buscar** del botón Inicio para encontrar la ruta del fichero Pbrush.exe. Crea un acceso directo desde el escritorio a este fichero. Cambia el nombre al acceso directo. Borra físicamente el acceso directo (para ello, comprueba la papelera si es necesario).

Crea una nueva carpeta en el escritorio. Llámala Mi Nueva Carpeta. Después de crearla, bórrala. Comprueba si se ha borrado realmente.

Abre alguna subcarpeta de la carpeta Windows. Activa el menú contextual de cualquiera de los iconos que hay en esa carpeta. Cierra la carpeta.

Consulta el contenido del submenú **Programas** (dentro del botón Inicio) en tu PC de prácticas. Prueba a ejecutar la aplicación **Bloc de notas** que se encuentra en **Accesorios**. Después ciérrala.

Ejecuta el programa **Bloc de notas** con la opción **Ejecutar** (el fichero Notepad.exe contiene este programa y suele encontrarse en el directorio Windows de C:) y ciérralo.

Busca todos los ficheros y carpetas cuyo nombre comience por not y cuya extensión (después del punto) tenga tres letras, siendo la segunda una x. Si has realizado bien la búsqueda, uno de los ficheros encontrados será Notepad.exe.

Abre algunas ventanas en el escritorio (abre un par de carpetas y ejecuta alguna aplicación). Coloca las ventanas primero en cascada, después en mosaico horizontal y, finalmente, en mosaico vertical. A continuación minimiza todas las ventanas y seguidamente restáuralas a su posición y forma anterior. Finalmente ciérralas todas.

Utilizando el explorador, crea una carpeta llamada Segundo Directorio dentro de C: y un acceso directo al **Bloc de notas**. Traslada el acceso directo al directorio Segundo Directorio. Por último, borra los elementos que has creado.

Si ya has acabado la sesión por hoy, apaga el ordenador utilizando la opción correspondiente. Acuérdate de apagar también la pantalla.

1.6. BIBLIOGRAFÍA

- Introducción. Microsoft WindowsXP. Manual del usuario (se adjunta con la licencia del programa).

- Temas de ayuda de Windows98 y WindowsXP.

CAPÍTULO 2

2. INTRODUCCIÓN AL ENTORNO KDE DE GNU/LINUX

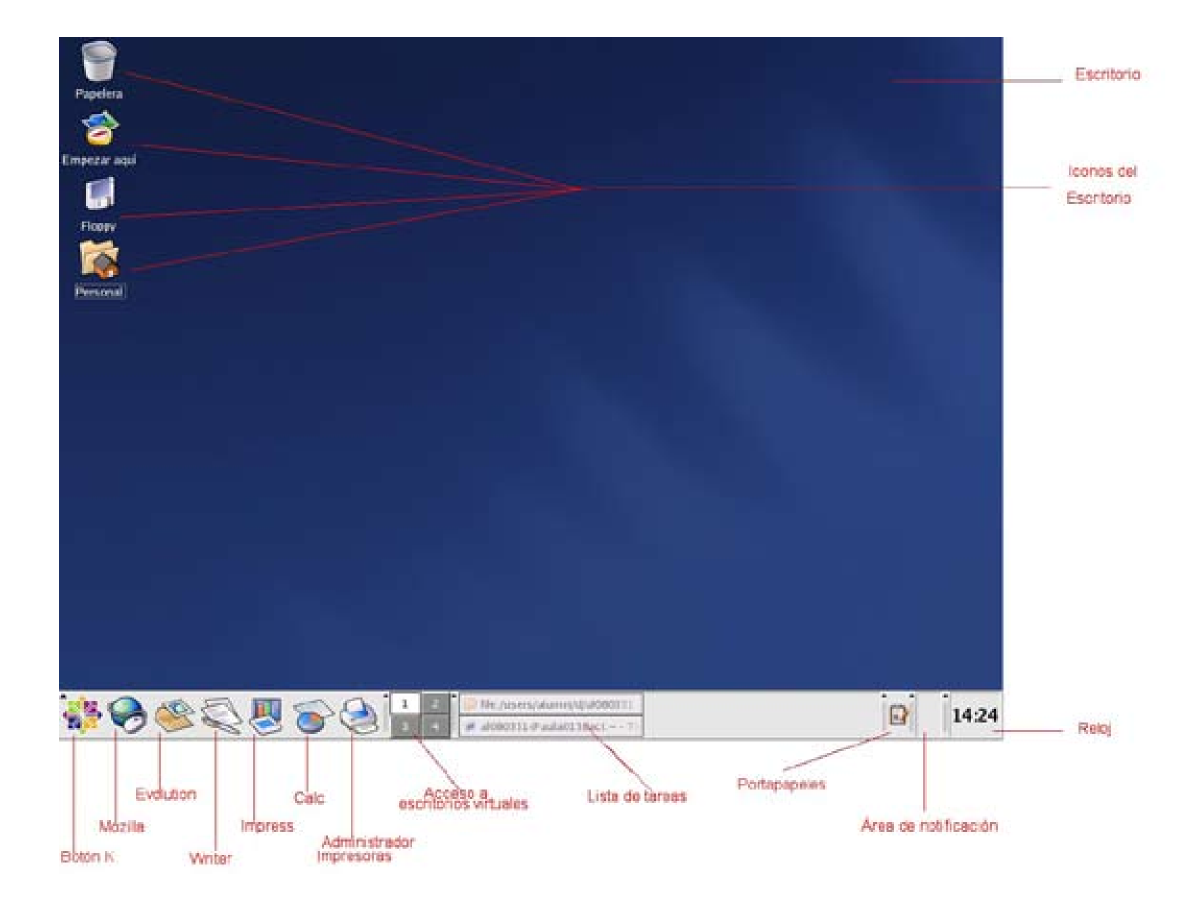
2.1. INTRODUCCIÓN Y OBJETIVOS

El sistema operativo que vamos a presentar en este cuadernillo es GNU/Linux. Este sistema operativo tiene como principal característica la de ser *libre*, esto es, se facilita el código fuente del sistema. Linux se distribuye bajo licencia GPL (General Public License) de GNU, de hecho resulta más adecuado hablar de GNU/Linux. Este sistema operativo es muy flexible y potente. La flexibilidad viene dada por su diseño modular y por la gran cantidad de aplicaciones de libre distribución que podemos encontrar: procesadores de texto, gráficos, compiladores, hojas de cálculo, gestores de ventanas, herramientas multimedia, herramientas para Internet, etc. La flexibilidad también viene dada por las posibilidades que se ofrecen a la hora de utilizar un entorno de trabajo determinado.

De esta forma, podemos trabajar en consola, en modo texto (de manera similar a como se hacía en MS-DOS), o bien utilizar una interfaz gráfica orientada a los usuarios, facilitando así el uso del ordenador mediante la incorporación de un entorno visual que intenta establecer una cierta correspondencia con aspectos relacionados con entornos de trabajo reales, admitiendo, junto con el teclado, el uso de periféricos apuntadores como el ratón. Gracias a todo ello se evita que el usuario tenga que aprender, recordar y, en determinados casos, teclear el nombre de las operaciones y órdenes que se desean dar al ordenador. Pero incluso aunque decidamos utilizar un entorno gráfico de trabajo, disponemos de diversas alternativas a considerar. De hecho es difícil encontrar entornos gráficos con el mismo aspecto para usuarios diferentes.

El entorno gráfico de Linux se conoce como sistema X Window, también conocido simplemente como "X". De este sistema ha habido multitud de versiones, pero no ha sido hasta la undécima versión, conocida como "X11", cuando se ha distribuido masivamente alcanzando la popularidad de la que disfruta hoy en día.

La filosofía general del sistema X Window (al igual que la de UNIX) es la de conseguir la funcionalidad global del sistema mediante la cooperación de componentes individuales, en lugar de depender de un único componente que realice todas las funciones. La ventaja de esta aproximación radica en que, llegado el caso, se puede modificar una determinada funcionalidad del sistema sin más que reemplazar únicamente el componente que la lleva a cabo. Desde el punto de vista del usuario final de aplicaciones, un sistema X consta de dos componentes principales:



- *Gestor de ventanas*: Se encarga de controlar la apariencia gráfica de las ventanas (poner la decoración en la barra de título, bordes, tamaño, comportamiento, etc.) y proporciona los medios necesarios para que el usuario pueda interactuar con las mismas (gestión de eventos de ratón y teclado, esto es, respuesta ante *clics* en determinadas zonas, arrastrar los bordes de las ventanas, combinación de teclas para "moverse" entre las ventanas activas, etc.). Existen varias alternativas: FVWM, AfterStep, Enlightenment, Sawfish, WindowMaker, ICEWM, etc.

- *Gestor o entorno de escritorio*: Proporciona un área de trabajo (el *escritorio*), barras de tareas para acceso rápido a las aplicaciones, gestión de ficheros y carpetas mediante iconos, funciones arrastrar y soltar (drag & drop), menús personalizables, temas de escritorio (personalización del entorno), múltiples escritorios virtuales, protector de pantalla, bloqueo de pantalla (con contraseña), etc. En la actualidad, los dos entornos de escritorio más populares en Linux son KDE y GNOME. En los ordenadores del aula de prácticas están instalados ambos. KDE viene configurado por defecto para usar su propio gestor de ventanas.

Este cuadernillo incluye una serie de ejercicios complementarios (destacados con un marco alrededor) que conviene realizar en el orden en que aparecen con el fin de facilitar la comprensión del mismo. El texto explicativo puede ser utilizado como apoyo para la realización de la práctica correspondiente. Ante cualquier duda, consulta con tu profesor de prácticas (tras la atenta lectura del material).

2.2. EL ENTORNO KDE

2.2.1. Principales elementos del escritorio KDE

■ Si has arrancado Linux y has introducido correctamente los datos de usuario (nombre y contraseña), al arrancar KDE observarás una imagen en la pantalla que debe ser parecida a la que se muestra en la Figura ?. Esta imagen del escritorio corresponde al entorno del curso 05/06. El de este año es *muy similar* y, puesto que las variaciones resultan mínimas, hemos decidido dejar esta imagen como referencia y comentar las diferencias en el texto siguiente.

■ KDE son las siglas de *K Desktop Environment* (entorno de escritorio K). Este entorno de escritorio tiene muchas semejanzas con los de Windows. Hasta hace unos años no cumplía estrictamente la licencia GPL, por lo que desde el proyecto GNU se impulsó la creación de una nueva herramienta para la gestión del escritorio en el entorno gráfico de Linux: GNOME. Los principales elementos que podemos apreciar en el escritorio son:

■ El *escritorio*, que es el entorno gráfico que utilizan usuario y ordenador para comunicarse. Ocupa toda la pantalla y sobre él se sitúan otros objetos.

■ Los *iconos* representan objetos a los que podemos acceder; por ejemplo, ficheros, carpetas (directorios), aplicaciones (programas), etc. Los podemos seleccionar y utilizar por medio del ratón. Una diferencia con respecto a Windows: en KDE hacer *clic* sobre un icono implica *"utilizarlo"*; si es un programa se ejecutará, si es un fichero de datos se visualizará usando el programa que esté definido a tal efecto (asociado). En la Figura ? podemos ver, empezando por la parte superior:

◆ La *papelera de reciclaje*. Podemos recuperar los ficheros que allí dejemos. Solamente los eliminaremos definitivamente cuando escojamos la opción **Vaciar la papelera** del menú que se despliega al hacer *clic* con el botón derecho del ratón sobre el icono que la representa.

◆ Un icono que permite acceder a la disquetera y, por ende, a los contenidos del disquete (montando su sistema de ficheros en el de Linux) si es que hay alguno introducido en ella.

◆ La carpeta *Personal*. Acceso a los contenidos del directorio personal del usuario (veáse la Figura ?). Si no la encuentras en el escritorio, entonces puede ser que aparezca al desplegar el menú del **botón K** (continúa leyendo). La carpeta *Desktop* contiene todo lo que aparece en el escritorio de KDE.

◆ En la versión de Linux para este curso, también encontrarás el icono *Sistema* (aunque no aparezca en la imagen anterior), que permite acceder, entre otras cosas, a los dispositivos de almacenamiento conectados al sistema, como por ejemplo un *pendrive* USB (veáse la Figuraque se muestra a continuación)



■ El *panel de herramientas*, desde el que se ejecutan la mayoría de las acciones. *Debe tenerse en cuenta que es posible que algunos de los elementos que se describen a continuación no aparezcan en la versión de este curso*. De izquierda a derecha nos encontramos con los siguientes elementos (iconos empotrados en el panel):

◆ **Botón K** (en el linux de aulas se utiliza un icono personalizado: el icono de CentOS): da acceso al menú KDE. Es el equivalente al botón **Inicio** en Windows XP.

◆ *Mozilla Firefox*: es un navegador WWW que usaremos más adelante en las prácticas de Internet. La Figura ?(b) muestra el aspecto que presenta. En realidad, este icono "apunta" al navegador predeterminado del sistema, que será *Firefox*.

■ *Gestor de correo (Evolution)*: gestor personal. Gestiona correo electrónico, agenda electrónica, libreta de direcciones, calendario, citas…. Es un programa con características y funcionamiento similar a *Outlook* de Microsoft, pero sin sus fallos de seguridad. En el entorno KDE se ejecuta *Kmail*, en el entorno Gnome *Evolution*.

■ *Writer* de OpenOffice: el procesador de textos de la *suite* ofimática libre más popular en Linux. Similar en cuanto a uso y funcionalidad a Microsoft *Word*.

■ *Impress* de OpenOffice: la herramienta para realizar presentaciones de la *suite* ofimática libre más popular en Linux. Similar en cuanto a uso y funcionalidad a Microsoft *Power Point*.

■ *Calc* de OpenOffice: la hoja de cálculo de la *suite* ofimática libre más popular en Linux. Similar en cuanto a uso y funcionalidad a Microsoft *Excel*.

■ *Escritorios virtuales*. Acceso a los cuatro escritorios virtuales que, por defecto, se muestran en KDE. Podemos definir más. Cada escritorio permite configurar su propio fondo, colores, etc. Cada uno de ellos gestiona las ventanas

> <u>Ejercicio 1:</u> Ejecuta el procesador de textos Writer de OpenOffice haciendo clic en el **botón K**, menú **Oficina**, opción **OpenOffice.org 2.4 Writer**.
>
> <u>Ejercicio 2</u>: Abre una ventana donde se muestren los contenidos del directorio inicial de trabajo del usuario. Para ello, haz *clic* en el icono *Personal* que está situado en el Escritorio o bien en el menú del **botón K**. El programa que muestra los contenidos es *Konqueror*.
>
> *Konqueror* es el navegador del entorno KDE. Está integrado en el sistema de ficheros de manera similar a como se integra *Internet Explorer* en Windows, de tal forma que podemos usarlo para "navegar" por él (utilizando su barra de herramientas, su cuadro de texto para direcciones, etc.). Proporciona una barra de menús, debajo hay una barra de herramientas (con iconos representando operaciones) y una caja de texto donde se pueden escribir tanto la ruta de la carpeta que deseas abrir, como una URL (dirección página web).

correspondientes a los programas que hemos lanzado ahí. Podemos mover ventanas de un escritorio a otro, datos de una ventana a la ventana de otro escritorio, etc. En la práctica permiten proporcionar cuatro entornos de trabajo distintos sobre el mismo escritorio. Para acceder a cada uno de los escritorios, basta con hacer *clic* en la pequeña área rectangular que los representa (cada uno se identifica mediante un número).

■ Cuadro *lista de tareas*. Permite acceder a las distintas ventanas que hay situadas en los diversos escritorios de KDE (maximizar, minimizar, cerrar, etc.).

■ El siguiente cuadro permite acceder a un programa para el acceso al portapapeles de KDE (zona de memoria donde se guarda el resultado de las operaciones cortar y copiar típicas del menú de edición de los programas).

■ A continuación, encontramos el *área de notificación*. Típicamente encontraremos "mensajes" de aplicaciones que indican eventos pendientes (tales como actualizaciones del sistema, etc.).

■ Finalmente encontramos el *reloj*. Muestra la hora y la fecha del sistema.

Siempre que quieras imprimir desde las aulas informáticas usarás la *pasarela WebPrint*. Ésta es la pasarela de impresión (a través del web) de la UJI. Mediante cualquier navegador WWW (*Firefox*, *Konqueror*, …) puedes enviar los documentos a los diferentes destinos de impresión. La Figura ?(a) muestra lo que ocurrirá si accedes a su dirección: `http://webprint.uji.es` (tras identificarte adecuadamente en el portal de la UJI). En realidad, esa imagen corresponde al curso 06/07; en la actualidad la pantalla que aparece es muy similar, aunque no es

exactamente la misma.

Ejercicio 3: Navega por el sistema de ficheros. Para abrir una carpeta o acceder a los contenidos de un fichero basta con hacer *clic* con el ratón sobre el icono que la/lo representa. Para volver atrás, pulsa el botón con forma de flecha hacia atrás. Para continuar hacia delante, has de hacer *clic* en el botón con forma de flecha hacia delante. El botón con forma de casita permite volver a la carpeta inicial de trabajo del usuario (Konqueror sobre el sistema de ficheros) o bien volver a la página web que hemos definido como página inicial a mostrar. El botón que hay al lado refresca/recarga los contenidos de la ventana (si estamos navegando por el sistema de ficheros, refresca los contenidos de la carpeta, apareciendo/desapareciendo los ficheros que hayamos creado/destruido; si es una página web vuelve a cargar sus contenidos). El botón con forma de X detiene la carga de contenidos (útil para páginas web). Además, el menú **Editar** contiene opciones que permiten realizar las operaciones de *cortar*, *copiar* y *pegar* con los elementos que hayamos seleccionado y en la carpeta donde estemos situados.

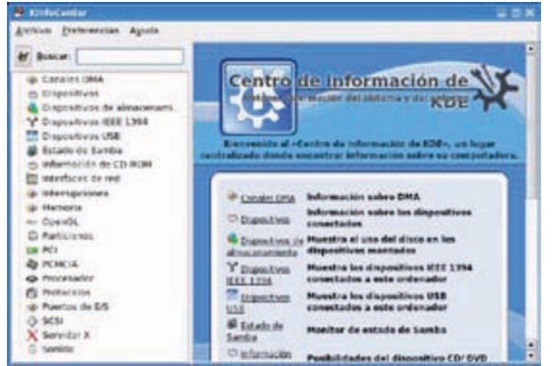
Cuando termines de navegar por el sistema de ficheros, haz *clic* en el botón de la barra de herramientas que te permite volver a mostrar los contenidos del directorio inicial de trabajo.

Ejercicio 4: Minimiza las dos ventanas que tienes ahora ocupando el escritorio. Para ello haz *clic* en los botones que las representan en el cuadro *lista de tareas* que hay en el panel. Vuelve a restaurarlas haciendo *clic* en dichos botones. Muévete al escritorio 2 y ejecuta *Calc* (**botón K**, menú **Oficina**, opción **OpenOffice.org 2.4 Calc**). Ve ahora al escritorio 3 y ejecuta *Mozilla Firefox*. De ahí pasa al escritorio 4 y ejecuta el programa *Terminal* que encontrarás dentro del menú **Herramientas del sistema**, accesible a través del **botón K**. Muévete por los escritorios. Observa, en cada uno de ellos, los contenidos del cuadro *lista de tareas*. Prueba, alternativamente, los efectos de hacer *clic* en cada uno de los botones de dicho cuadro. Finalmente restaura todas las ventanas, cada una en su escritorio y vuelve al escritorio 1

A continuación, quisiéramos comentar un par de cuestiones más sobre KDE: una, acerca del *Centro de Información*; otra, acerca del *Centro de Control*. Para ejecutar el primero tendrás que desplegar el menú del **botón K**: lo encontrarás dentro del menú **Herramientas del sistema**, opción **KInfoCenter**. El segundo tendrás que ejecutarlo mediante la opción **Ejecutar comando...** del menú que se despliega al pulsar el **botón K**: tienes que teclear **kcontrol** en la ventana que aparece y hacer *clic* en el botón **Ejecutar**.

Las figura que se muestra a continuación ilustra diversos aspectos del *Centro de Información* de KDE. En la parte izquierda de la ventana aparece un panel que contiene diversas secciones referentes a los dispositivos y la configuración de nuestro sistema . Por defecto, el *Centro de Información* muestra este índice de secciones. Podemos consultar diversa información acerca de configuración del sistema y características del hardware instalado: dispositivos USB, SCSI, ocupación de la memoria RAM, número y tamaño de las particiones de disco, puertos de

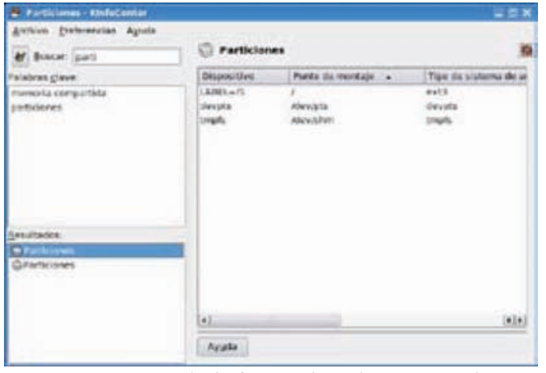
Entrada/Salida, información de la tarjeta gráfica (X Window), etc. Para acceder a una información concreta basta con hacer *clic* en la opción correspondiente de la parte izquierda. Dicha información se visualizará en la parte derecha de la ventana.



a.- Ventana principal del Centro de Información.



b.- Usando el cuadro Buscar del Centro de Información



c.- Se muestra la información de una sección.



d.- Haciendo clic en el botón Ayuda accedemos a la ayuda de KDE

El *cuadro Buscar* (sito en la parte de arriba del panel de la izquierda) permite realizar búsquedas de información sobre el sistema y el hardware empleando palabras clave. Según comenzamos a escribir aparece una lista elaborada de palabras clave en la parte superior (*Palabras clave:*) (figura *b*). Los resultados hallados se muestran abajo (*Resultados:*). La parte *(c)* de la figura muestra el acceso a uno de los resultados proporcionados. Si hacemos *clic* en el icono de la derecha (con forma de salvavidas) tendremos acceso a la ayuda contextual. Haciendo *clic* en el *botón Ayuda* accederemos a la sección correspondiente del manual de KDE (veáse la parte *(d)* de la figura de la página anterior, que muestra el acceso a un tema de ayuda de KDE). Siempre que hagamos *clic* en el icono que hay a la izquierda del *cuadro Buscar*, el panel de la parte izquierda volverá a mostrar el índice de secciones tal como aparece en la parte *(a)* de la Figura.

El *Centro de Control* de KDE da acceso a diversas opciones de configuración del sistema y del escritorio: conexión a Internet, modificar fecha y hora del sistema, teclado, carga de módulos del kernel, ratón, impresoras, sonido, gestión de usuarios, aspecto del escritorio, comportamiento de KDE, navegación web, etc. Muchas de estas opciones no las podremos ejecutar ya que no está permitido hacer modificaciones a un usuario “normal”. Para poder utilizarlas es necesario ser el usuario *root* (administrador del sistema). Podremos acceder a ellas a través del panel

de la izquierda (veáse la figura que se muestra a continuación). Tal como ocurría con el *Centro de Información*, el *cuadro Buscar* y el *botón Ayuda* permiten, respectivamente, acceder a las herramientas de configuración deseadas mediante la utilización de palabras clave y buscar ayuda relacionada con las herramientas del *Centro de Control*.



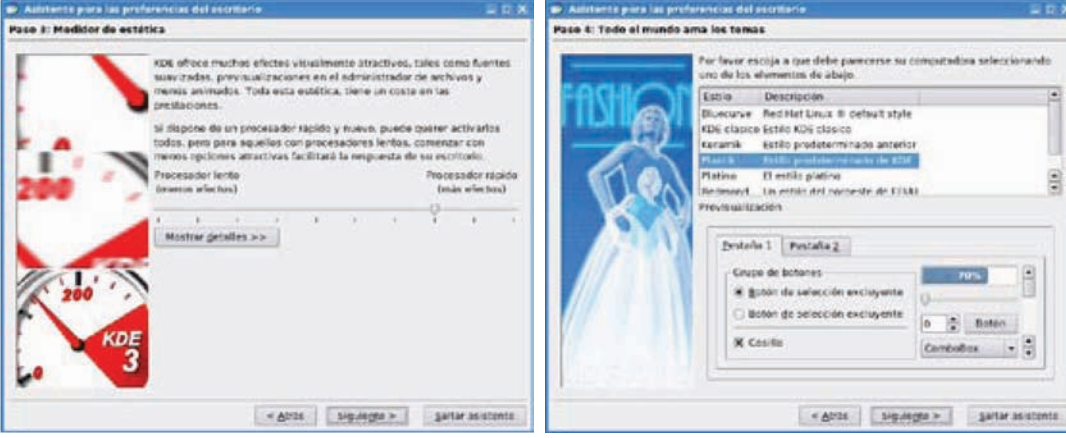
Finalmente, queremos explicarte brevemente cómo puedes realizar la *personalización* del escritorio KDE en cinco pasos. Utilizando el *Asistente para las preferencias del escritorio* podrás configurar de una forma fácil y rápida los siguientes aspectos: idioma, comportamiento del gestor de ventanas, teclado, comportamiento del escritorio, "efectos especiales" (animaciones, previsualizaciones, etc.) y decoración de las ventanas y del escritorio. Es posible que ni el comportamiento ni el "aspecto" de KDE al arrancar se ajuste a lo visto en las clases de teoría y a lo explicado en este material de apoyo

Esto se debe a que *CentOS* utiliza un *tema propio* de KDE que lo asemeja en comportamiento a GNOME, el otro gestor de escritorio popular en el mundo Linux. Así, por ejemplo, habrás observado que hay que hacer *doble clic* para ejecutar programas y abrir carpetas y ficheros en lugar de un único *clic* como habíamos explicado. Para conseguir un aspecto y un comportamiento más cercano al estándar de KDE es necesario que ejecutemos el *Asistente para las preferencias del escritorio* y escojamos, simplemente, las opciones por defecto. Para ejecutar este asistente tendrás que hacer *clic* en el **botón K**, seleccionar la opción **Ejecutar comando...**, teclear **kpersonalizer** en la ventana que aparece y hacer *clic* en el botón **Ejecutar**. La figura que se muestra continuación ilustra las elecciones que debes realizar en cada uno de los pasos de la configuración de KDE. Éstas son las que aparecen por defecto (excepto en el paso 2, en el que debes elegir previamente la opción que muestra la imagen), con lo cual basta con ir pulsando el botón **Siguiente**.



a.-Escogiendo país e idioma.

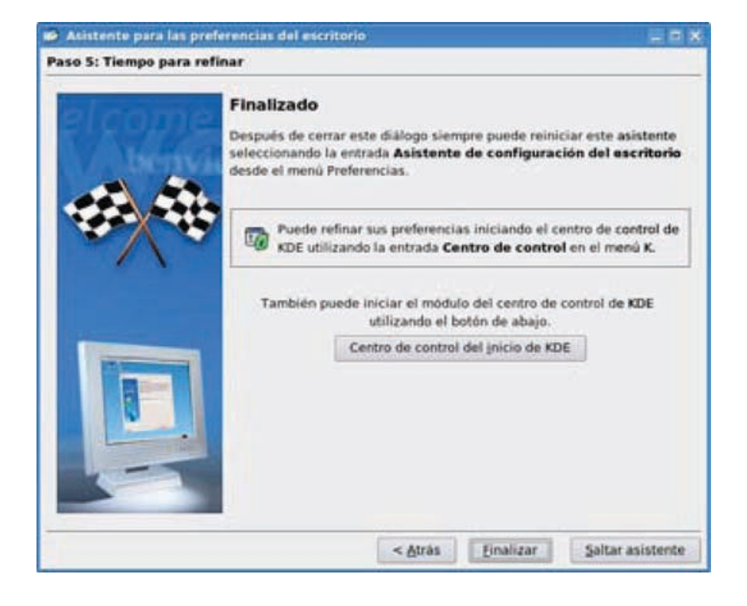
b.-Configurando el comportamiento del gestor de ventanas.



c.- Definiendo el nivel de ``efectos especiales" del escritorio

d.- Eligiendo decoración de ventanas y aspecto de sus botones

Al finalizar el proceso, aparecerá la siguiente ventana. En ese momento debes *clic* en **Finalizar**.



Ejercicio 5.- Ejecuta el \textsl{Asistente para las preferencias del escritorio. Personaliza el entorno KDE tal cual se indica en las figurasanteriores. Éste será el aspecto y comportamiento que tendrá de ahora en adelante tu entorno KDE.

2.2.2. Manejo básico de ventanas. Escritorios virtuales

La mayoría de los *objetos* que podemos encontrar en el entorno KDE tienen asociado su propio menú contextual. Éste funciona de manera similar al de Windows. Se activa haciendo *clic* con el botón derecho del ratón sobre el objeto en cuestión

2.2.2.1. El escritorio

Si hacemos *clic* con el botón derecho del ratón en cualquier superficie libre del escritorio, emergerá un menú contextual parecido al que muestra la figura de la izquierda. En él podemos apreciar opciones para terminar la sesión KDE, bloquear la pantalla, configurar el fondo del escritorio, organizar y alinear los iconos, etc.

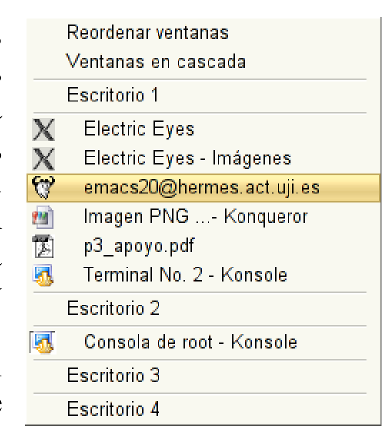


Las opciones **Ventanas en cascada** y **Reordenar ventanas** (dentro del menú **Ventanas**) permiten posicionar las ventanas de las aplicaciones de forma que se pueda acceder a todas ellas en el escritorio (que ninguna quede completamente oculta). La opción **Ejecutar comando…** permite ejecutar un programa invocando su nombre.

La opción **Iconos** presenta un menú que permite, entre otras cosas, **Alinear horizontalmente** y **Alinear verticalmente** los iconos del escritorio considerando la pantalla como si fuese una rejilla compuesta por filas y columnas, mientras que la opción **Ordenar iconos** los dispone en orden secuencial (uno debajo de otro empezando por la izquierda y pasando a la siguiente hilera cuando se acaba la pantalla por abajo) según el orden que seleccionemos: por nombre, tamaño, tipo, ….

Ejercicio 6: Sitúate en el escritorio 1. Allí tendrás dos ventanas: las correspondientes a *Writer* y *Konqueror* (éste mostrando los contenidos del directorio personal). Activa el menú contextual del escritorio y ejecuta las opciones **Reordenar ventanas** y **Ventanas en cascada**. Observa qué ocurre. Ejecuta ahora la opción Ejecutar comando…, escribe *ark* y pulsa Enter o haz clic en el botón **Ejecutar**. A continuación, arrastra con el ratón (haz clic con el botón izquierdo y sin soltar arrastra el icono) los iconos por el escritorio, desordenándolos. Prueba el efecto de las opciones **Alinear horizontalmente** y **Alinear verticalmente** del menú que despliega la opción **Iconos**. Ahora prueba a **Ordenar iconos** por **nombre**, **tamaño** y **tipo**. Finalmente, minimiza y restaura todas las ventanas haciendo clic sobre el icono que las representa en el cuadro *lista de tareas*.

2.2.2.2. El cuadro lista de tareas



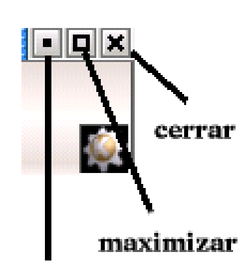
Como ya hemos visto, el cuadro lista de tareas proporciona mecanismos para manejar las ventanas de las aplicaciones que tenemos abiertas en cada escritorio virtual. Si las ventanas están minimizadas, al hacer clic sobre el icono que las representa en el cuadro se restauran a sus dimensiones y posición originales; si no se "activan" (esto es su ventana pasa a primer plano, recibiendo los eventos de ratón y teclado).

Aún más, si haces clic con el botón central del ratón en una zona despejada del escritorio, verás que se despliega un menú similar al de la derecha. En él aparecerán, debajo del título de cada escritorio, la lista de aplicaciones abiertas en el susodicho. Además también podemos apreciar las opciones **Reordenar ventanas** y **Ventanas** en cascada con el mismo significado que en el menú contextual del escritorio. Otra posibilidad para cambiar rápidamente entre las ventanas correspondientes a las aplicaciones abiertas en un escritorio es usar la combinación de teclas Alt + Tab.

Si haces clic con el botón derecho del ratón encima de alguno de los iconos que representa una ventana abierta en el escritorio, aparecerá un menú contextual que te permitirá, entre otras cosas, minimizar, maximizar o restaurar la ventana y mover la ventana a otro escritorio (**Al escritorio**).

> Ejercicio 7: Utiliza la lista de tareas (bien el menú al que se accede haciendo clic con el botón central del ratón en el escritorio, bien los iconos) para pasar de una a otra de las aplicaciones que tienes en ejecución. Usa, igualmente, las teclas Alt + Tab. Finalmente, vuelve al escritorio 1 y prueba las opciones Reordenar ventanas y Ventanas en cascada.
>
> Ejercicio 8: Sitúate en el escritorio 1. Ejecuta el menú contextual asociado al icono que representa la ventana donde Konqueror está mostrando los contenidos de tu carpeta personal. Minimízala. Vuelve a activar el menú contextual. Restaura la ventana a su posición y tamaño original.

2.2.2.3. Las ventanas



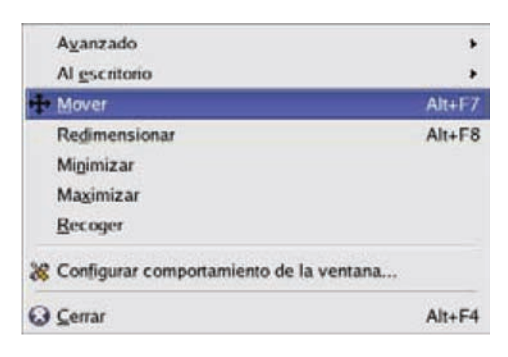
Cada aplicación que se está ejecutando en el entorno KDE tiene asociada una ventana en la que muestra información y pide datos al usuario. Cada ventana es independiente de las demás. En cada momento, hay una ventana distinguida que es la que recibe las pulsaciones de teclado y los clics del ratón del usuario. Se dice que esa ventana tiene el foco. Dicha ventana se puede identificar porque tiene un color distinto en su barra de título que, en el caso de los ordenadores del aula, es azul (mientras que el resto de las ventanas tiene un color grisáceo). Además, podemos apreciar que en el cuadro de la lista de tareas el icono que

representa dicha ventana aparece como “hundido” (en un tono más opaco), mientras que los otros aparecen más iluminados.

En la parte derecha de la barra de título de todas las ventanas aparecen 3 botones. Cada uno de ellos permite realizar una acción determinada sobre la ventana, por este orden minimizar, maximizar/restaurar y cerrar la ventana.

El botón minimizar hace que la ventana desaparezca del escritorio pasando a estado latente. Se dice que la aplicación está minimizada y, en ese estado, el usuario no puede enviarle datos. También podemos acceder a estas opciones, haciendo clic en el icono que aparece en la parte izquierda de la barra de título. Este icono no es el mismo en todas las ventanas, ya que varía en función de la aplicación como podéis observar (si es que no lo habéis observado ya).

Si hacéis doble clic en la barra de título de una ventana, ésta se enrolla (como si fuese una persiana), esto es desaparece el “cuerpo” de la ventana y solamente queda visible la barra de título. Esto puede ser útil si queremos despejar un poco el escritorio. Para desenrrollar la persiana basta con volver a hacer doble clic sobre la barra de título. Para mover la ventana, esto es desplazarla por el escritorio, hemos de hacer clic en la barra de título y, sin soltar el botón izquierdo del ratón, desplazar el mismo hasta que la ventana alcance la posición deseada en el escritorio. En ese momento debemos soltar el botón del ratón.



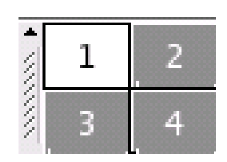
Para cambiar el tamaño de una ventana podemos utilizar el menú que emerge al hacer clic en el icono que aparece en la parte izquierda de la barra de título (opción Redimensionar).

Seguramente nos resultará más cómodo utilizar las “tiras” que hay en los esquinas inferiores (izquierda y derecha) de las ventanas. Arrastrando y soltando con el ratón podemos fijar las dimensiones deseadas a las ventanas.

Finalmente, si haces clic con el botón derecho del ratón sobre la barra de título de una ventana, emergerá el menú contextual de la misma. Dicho menú contextual es idéntico al que se muestra en la figura de la izquierda. Resulta interesante la opción Al escritorio, que permite enviar la ventana a algún otro escritorio virtual.

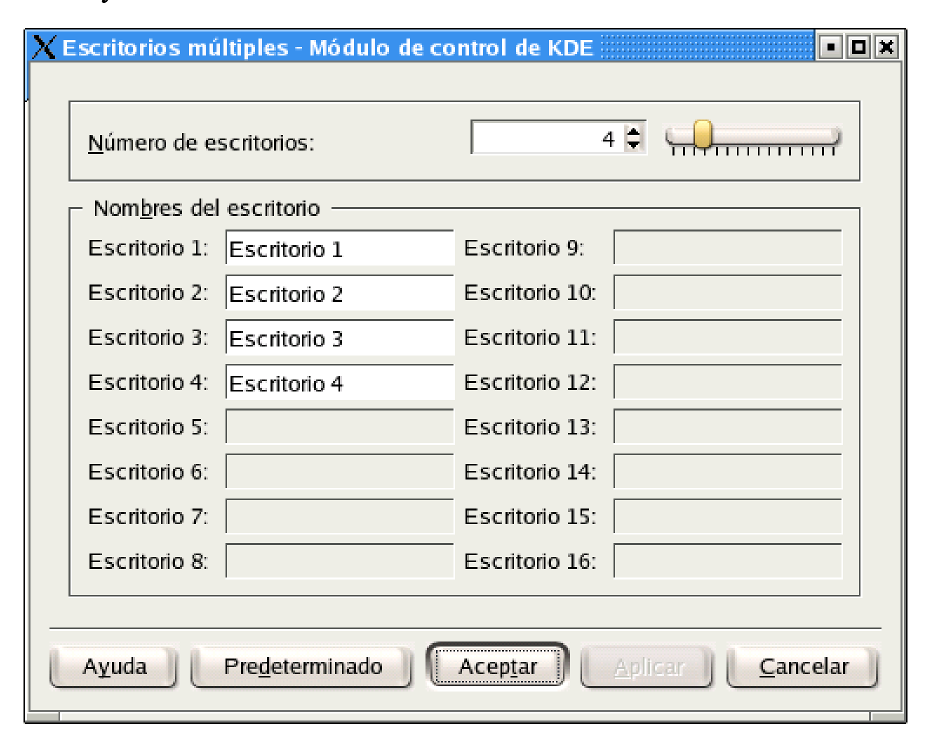
Ejercicio 9: Muévete por los escritorios. “Juega” con las ventanas de las aplicaciones que tienes abiertas: minimízalas, maximízalas, restaúralas, cámbiales el tamaño, pásalas de un escritorio a otro, enróllalas, desenróllalas, …, prueba todas las opciones que hemos visto. Utiliza distintas formas de realizar las operaciones (bien el menú del icono de la parte izquierda de la barra de título, bien los botones de la parte derecha) en los casos en los que puedas elegir. Para finalizar este ejercicio, cierra todas las ventanas que habías abierto anteriormente (en todos los escritorios). Todos los escritorios deben de quedar, pues, vacíos.

2.2.2.4. Escritorios virtuales



Para acceder a los cuatro escritorios virtuales que, por defecto, utiliza KDE, hemos de usar el espacio del panel de herramientas que aparece en la Figura ?. Si hacemos *clic* en cualquiera de las áreas rectangulares que indican la presencia de un escritorio (numeradas del 1 al 4) accedemos a dicho escritorio. Allí podremos manipular las ventanas de las aplicaciones que hemos lanzado desde dicho escritorio. La combinación de teclas Ctrl + Tab permite cambiar de un escritorio a otro.

Si hacemos *clic* en cualquier punto de este recuadro con el botón derecho del ratón, accedemos al menú contextual del mismo. En él aparece, entre otras, la opción **Configurar escritorios ...**, que muestra la ventana que ilustra la figura qye se myuestra en la página siguiente. En esta ventana puedo cambiar el nombre de los escritorios y aumentar o disminuir la cantidad de los mismos.



Para cambiar el nombre de un escritorio, basta con situarse en el recuadro de texto correspondiente, borrar el nombre anterior y escribir el nuevo. Para aumentar/disminuir la cantidad de escritorios virtuales basta con mover hacia delante/atrás la ruedecita que aparece en la parte superior de la ventana (**Número de escritorios**). Para mover la ruedecita haz *clic* sobre ella con el ratón y sin soltar el botón izquierdo arrástrala hacia delante/atrás. Para conservar definitivamente los cambios realizados haz *clic* en el botón **Aceptar**. Si sólo quieres ver el efecto haz *clic* en **Aplicar**. Si te arrepientes, pulsa **Cancelar**.

Ejercicio 10: Utilizando el menú contextual del recuadro que muestra los escritorios virtuales en el panel de herramientas, cambia a modo de visualización por nombre (opción **Nombre de escritorio** dentro de **Opciones del paginador**). Vuelve a cambiar a número (**Número de escritorio**). Accede a la opción **Configurar escritorios virtuales** y cambia el nombre de los escritorios.

Vuelve al modo de visualización por nombre. Crea un escritorio virtual más. Llámalo como quieras. Observa el efecto. Accede al nuevo escritorio que has creado. Arranca Writer y Calc. Minimiza las ventanas. Muévelas. Observa los contenidos de la lista de tareas (haz clic en el icono con forma de punta de flecha hacia arriba). Cierra ambas ventanas. Vuelve al escritorio 1. Elimina el escritorio virtual que acabas de crear (a través de **Configurar escritorios virtuales**). Vuelve al modo de visualización por número.

2.2.3. El botón K

El botón K agrupa mediante menús (la mayoría de) las diversas aplicaciones que hay instaladas en nuestro sistema GNU/Linux. En la parte superior aparece la lista de las aplicaciones recientemente utilizadas. A continuación, vienen los diversos menús que agrupan aplicaciones de forma lógica: por ejemplo el menú Internet agrupa una serie de herramientas para acceso/utilización de diversos servicios de Internet. Algunos de los menús destacables, que se ilustran en las figuras que se muestran a continuación (¡ojo! , las imágenes empleadas corresponden al curso 06/07) y que encontrarás también este curso, son:

- Gráficos: contiene diversas aplicaciones para la visualización de documentos e imágenes (p.e. KPDF para visualizar documentos en formato PDF), edición y manipulación de imágenes (p.e. el GIMP), utilidades para escanear documentos e imágenes, edición y captura de fotos desde cámara digital, etc.

- Internet: contiene varias aplicaciones para utilizar servicios de la denominada “red mundial de ordenadores”: correo electrónico, navegador WWW, mensajería instantánea, transferencia de ficheros (FTP), etc. Algunos de ellos los utilizaremos en las prácticas de Internet.

- Oficina: permite acceder, entre otras, a las herramientas que componen el paquete ofimático OpenOffice 2.4: procesador de textos, hoja de cálculo, programa para presentaciones, etc. En una práctica posterior trabajaremos con el procesador de textos Writer incluido en esta suite ofimática.

Además, tenemos las opciones: **Ejecutar comando**, que muestra una ventana donde podemos teclear el nombre del programa que deseamos ejecutar (como ya hemos visto anteriormente) y **Bloquear sesión** y **Terminar**…, que permiten, respectivamente, arrancar el protector de pantalla protegido con contraseña y terminar la sesión de trabajo KDE.



Es posible que haya algún programa instalado en el sistema Linux que no sea accesible a través de los menús de KDE. En ese caso, podremos ejecutarlo abriendo una consola X (o a través de **Ejecutar comando**), escribiendo su nombre (realmente el nombre del fichero que lo contiene) y pulsando la tecla Enter. En algunas ocasiones (si el sistema no ha sido adecuadamente configurado) es posible que tengamos que escribir la ruta completa del fichero que contiene el programa.

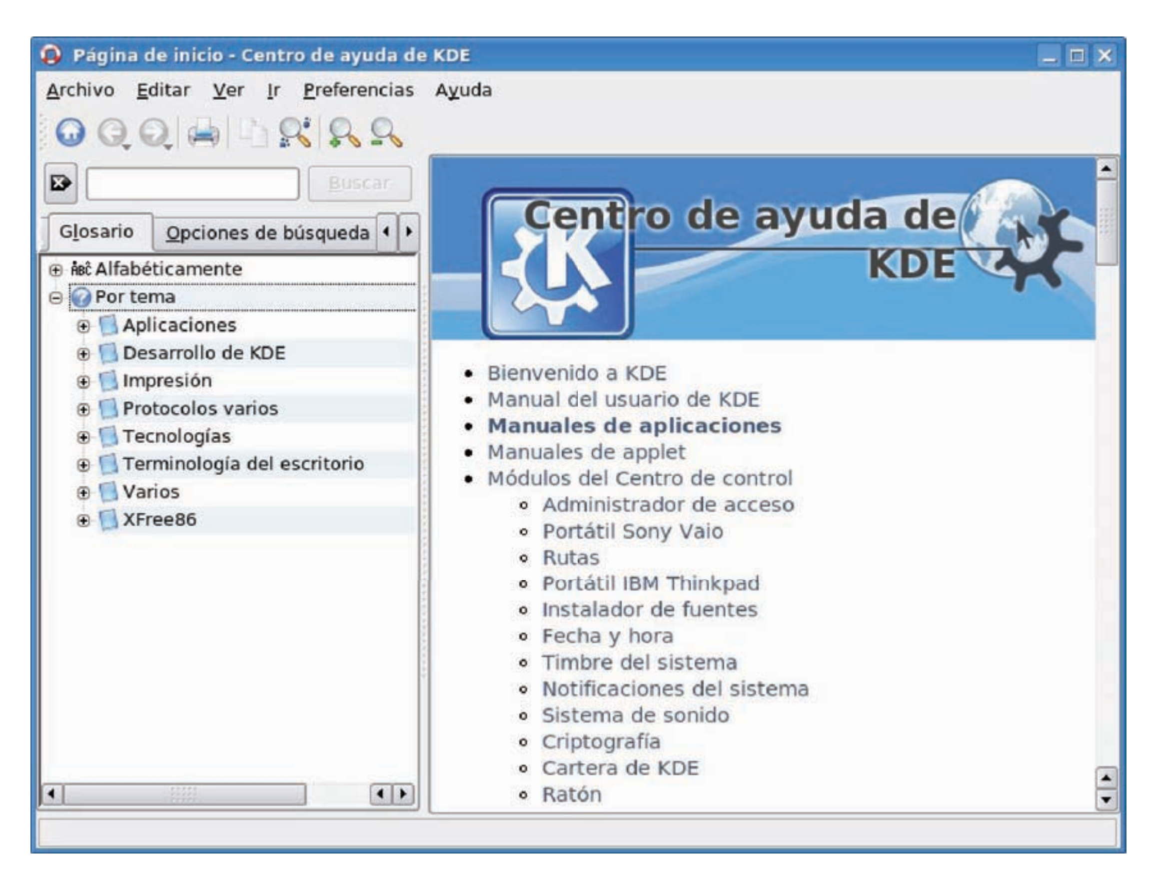
2.2.4. Uso de la ayuda

Para acceder a la ayuda del entorno KDE tendremos que desplegar el menú del **botón K**, seleccionar la opción **Ejecutar comando...**, teclear **khelpcenter** en la ventana que aparece y hacer *clic* en **Ejecutar**. Si ejecutamos la ayuda aparecerá una ventana similar a la que muestra la Figura ?.

En la parte izquierda de la ventana tenemos, en la *ficha Glosario*, la visión de las diferentes carpetas y temas de ayuda de KDE (bien ordenados alfabéticamente o por tema). Si hacemos *clic* sobre los iconos con forma de *libro cerrado*, se desplegarán los diversas temas de ayuda de que consta. Si hacemos *clic* sobre un icono en forma de *documento*, en la parte derecha se mostrará el contenido del tema de ayuda que contiene. Por defecto, se muestra el contenido de la ficha *Centro de ayuda de KDE*. Existen varios temas de ayuda, tanto para usuarios novatos de KDE y Linux en general, como para usuarios ya “veteranos” en el manejo de GNU/Linux.

Sería interesante que leyeses los temas *Manual del usuario de KDE* y los correspondientes a los *Tutoriales* y *Manuales de aplicaciones* que aparecen en la parte derecha de la ventana inicial (más abajo). También resultan instructivos los temas de ayuda sobre la utilización y opciones de configuración del *Centro de Control* KDE, *Módulos del Centro de control*, y sobre el Centro de Información,

Módulos del KinfoCenter.



2.2.5. Konqueror: gestión de ficheros y acceso a WWW

En resumen, *Konqueror* es a KDE lo que *Internet Explorer* es a Windows. *Konqueror* es, básicamente, un navegador WWW integrado en el sistema de ficheros, de tal forma que permite "navegar" por el mismo, mostrando los contenidos de los ficheros (invocando internamente las aplicaciones a las que éstos están asociadas) y de las carpetas. Para abrir una carpeta (acceder a sus contenidos) basta con hacer *un único clic* en el icono que la representa. Lo mismo para acceder al contenido de un fichero.



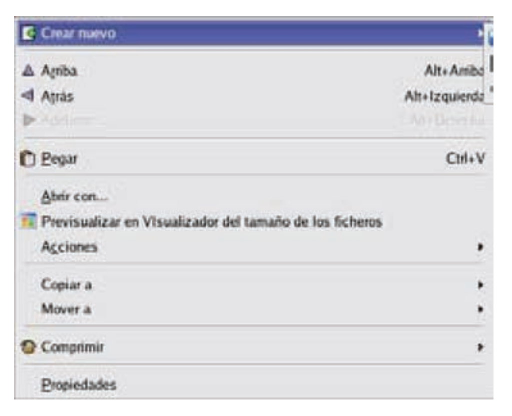
Podemos usar los botones de la barra de navegación para navegar por el sistema de ficheros tal como se comentaba en el Ejercicio 3. Allí se mostraba el funcionamiento de algunos de los botones (los que más interesan a nuestros propósitos introductorios) de la barra de navegación de *Konqueror*. No lo volveremos a repetir ahora. Consulta el texto de dicho ejercicio para refrescar la memoria.

Para seleccionar un fichero o grupo de ficheros, podemos hacer *clic* con el botón izquierdo del ratón en la esquina superior izquierda de lo que sería el área rectangular imaginaria que cubre los elementos a seleccionar y sin soltar el botón, arrastrar el ratón

hasta cubrir todos los elementos que queremos seleccionar. También puedes seleccionar un fichero haciendo *clic* en el icono que lo representa al mismo tiempo que pulsas la tecla claroControl. Luego, puedes extender la selección de dos maneras: seleccionando una serie de elementos consecutivos haciendo *clic* en el icono del fichero o carpeta que ocupa la última posición en la serie al mismo tiempo que pulsas la tecla claroShift; seleccionando elementos alternos haciendo *clic* sobre el icono del elemento que deseas añadir a la selección al mismo tiempo que pulsas la tecla claroControl.

Para copiar, cortar y pegar puedes utilizar los *botones* que, a tal efecto, existen en la barra de navegación, o también las correspondientes opciones (**Copiar**, **Cortar**, **Pegar**) del menú **Editar** en la barra de menús. Para copiar ficheros y/o carpetas, deberás seleccionar los iconos que representan a los elementos que deseas copiar; a continuación, tendrás que hacer *clic* en el botón para copiar o bien en la opción **Copiar** del menú **Editar**. Finalmente, tendrías que situarte en la carpeta donde quieres copiar los ficheros y allí harías *clic* en el botón para pegar (consulta el Ejercicio 3) o bien en la opción **Pegar** del menú **Editar**. Para mover los ficheros de una carpeta a otra tendrías que hacer lo mismo, salvo que una vez hayas seleccionado los elementos a trasladar tendrías que hacer *clic* en el botón de cortar o bien seleccionar la opción **Cortar** del menú **Editar**.

Estas operaciones también pueden ser realizadas usando el menú contextual (ya sabes, botón derecho del ratón) de los iconos (veáse la Figura ?). Si tienes más de un icono seleccionado basta con que uses el menú contextual de uno de ellos; la operación se realizará sobre todos los elementos seleccionados. En este menú contextual destacan otras opciones como **Deshacer :** la última operación efectuada (si se puede) (a la que también se puede acceder desde la opción **Deshacer:** del menú **Editar** de *Konqueror*), **Renombrar** para cambiar el nombre de un fichero o carpeta, **Mover a la papelera** para mover los elementos a la papelera, **Eliminar** para borrar los elementos seleccionados y **Propiedades** que muestra en una ventana las propiedades del elemento o elementos seleccionados (tamaño, permisos, etc.)

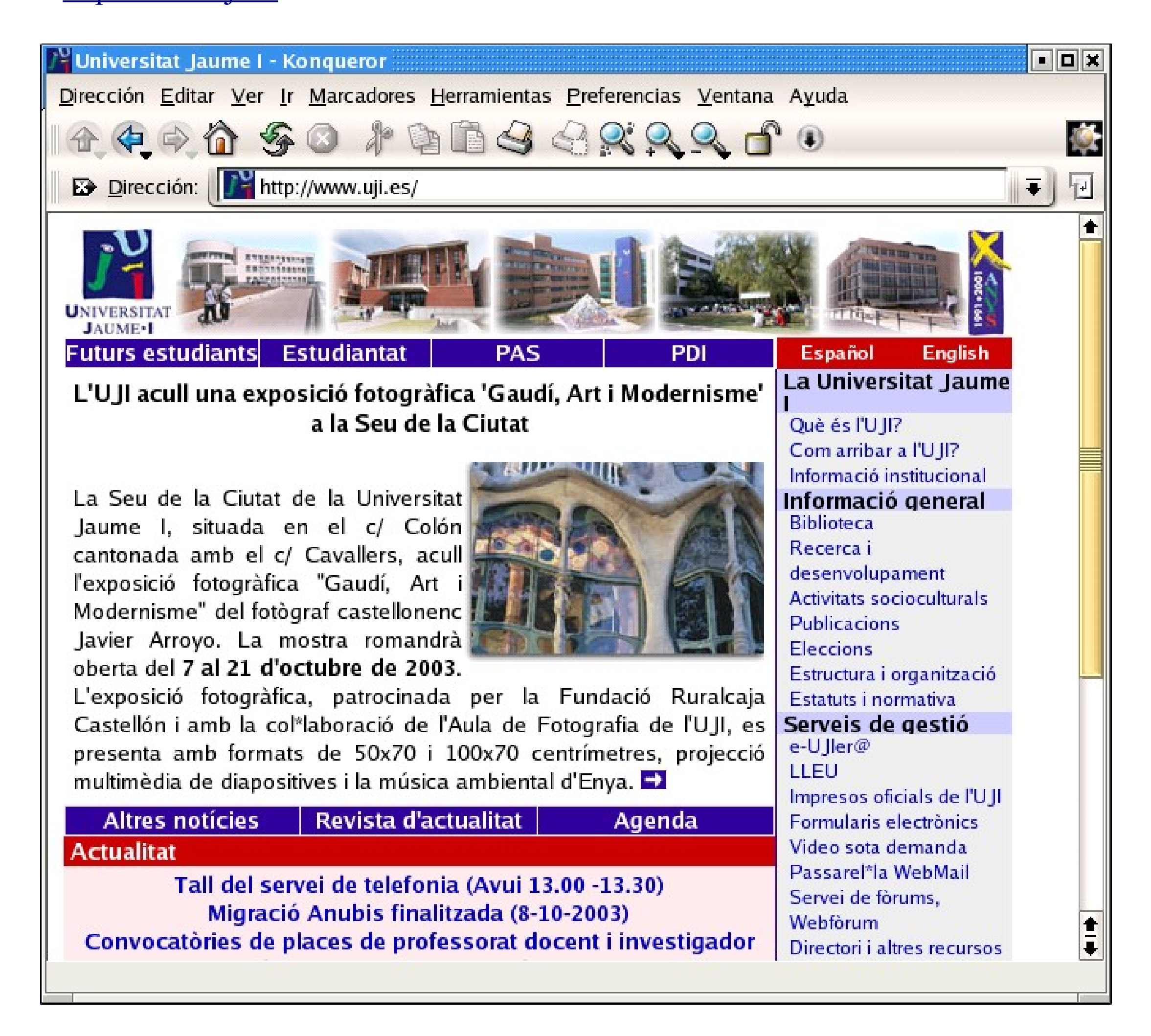


Konqueror tiene también su propio menú contextual. Es el que se muestra en la figura de la izquierda. En él destacan las opciones **Copiar a** y **Mover a** que permiten, respectivamente, copiar y trasladar fácilmente los objetos seleccionados en la carpeta actual, **Crear Nuevo** que permite crear nuevas carpetas y ficheros en una ventana determinada y **Comprimir** que permite empaquetar y comprimir en un nuevo archivo (o añadir a uno ya existente) todo el contenido del directorio que actualmente está mostrando *Konqueror*.

Hasta hace poco, estábamos acostumbrados a navegar (tanto por Internet como por el sistema de ficheros) empleando varias ventanas para acceder a varios sitios al mismo tiempo. Esto resultaba incómodo, ya que enseguida llenábamos el escritorio con ventanas del navegador. Ahora podemos visualizar el contenido de varios sitios empleando *una única ventana del navegador*. ¿Cómo? Fácil, empleando una *pestaña* para visualizar un sitio determinado. Y todas las *pestañas* se agrupan en la

misma ventana del navegador. Para abrir una nueva pestaña tan sólo tenemos que elegir la opción **Pestaña nueva** en el menú **Dirección**. Recuerda igualmente que, para mover carpetas y/o ficheros a una determinada carpeta destino también podemos arrastrar y soltar (tal como se hacía en Windows) los elementos seleccionados en la ventana actual en la ventana correspondiente a la carpeta destino.

Konqueror también es un navegador WWW. Para arrancarlo como navegador basta con desplegar el menú del **botón K** y acceder a la opción **Internet**, **Konqueror**. En la caja de la parte superior, tendríamos que teclear entonces una URL (la dirección de una página web) para visualizar sus contenidos y navegar por Internet. Otra posibilidad, tal como hacíamos en Windows con *Explorer*, es usar la propia ventana "del sistema de ficheros" (la que aparece cuando hacemos *clic* en el icono *Personal* que hay en el escritorio) para navegar por Internet, tecleando la URL deseada en la caja que aparece en la parte superior de la ventana. La siguiente Figura ilustra el aspecto de Konqueror al mostrar los contenidos de la página web de la UJI: <http://www.uji.es>.



Ejercicio 12: Utilizando Konqueror, crea una carpeta denominada Prac en tu directorio personal. Abre la carpeta y crea otras dos carpetas en su interior: una denominada Cosas Web y otra denominada Mis Cosas.

Ejercicio 13: Ejecuta Writer de OpenOffice (recuerda que está en el menú **Oficina** del botón **K**). Utiliza Konqueror para conectarte a la página web de la UJI: <http://www.uji.es>. Copia algunas líneas del texto de dicha página web en la ventana de Writer. Guarda el fichero (**Archivo Guardar**) con el nombre Uji (automáticamente Writer le pondrá la extensión .odt) en la carpeta Cosas Web. Crea un nuevo fichero en Writer (**Archivo Nuevo**). Conéctate ahora a la web del periódico El Mundo: <http://www.elmundo.es>. Nuevamente copia algunas líneas en la ventana de Writer y guarda el fichero resultante en la carpeta Cosas Web con el nombre Mundo. Procede de la misma manera con la web del periódico El País: <http://www.elpais.es>. El fichero que crees lo salvarás igualmente en la carpeta Cosas Web con el nombre Pais.

Ejercicio 14: Crea un nuevo fichero con Writer. Copia los dos primeros párrafos de esta sección. Guarda el fichero creado con el nombre Cosa1 en la carpeta Mis Cosas. Crea otro nuevo fichero con Writer y ahora copia el texto correspondiente al ejercicio anterior. Guarda el fichero resultante con el nombre Cosa2 en la carpeta Mis Cosas. Finalmente, termina la ejecución de Writer (**Archivo Terminar**).

Ejercicio 15: Copia la carpeta Mis Cosas y su contenido a otra carpeta que habrás de crear en tu directorio personal y que se llamará Cosas-repe.

Ejercicio 16: Mueve la carpeta Cosas-repe y todo su contenido dentro de la carpeta Prac. Observa el resultado.

Ejercicio 17: Deshaz la acción que acabas de realizar (usa **Deshacer**). Observa los resultados. Vuelve a repetir la acción del ejercicio anterior.

Ejercicio 18: Cambia el nombre de los dos ficheros contenidos en la carpeta Cosas-repe a Cosa1-repe.odt y Cosa2-repe.odt respectivamente.

Ejercicio 19: Borra la carpeta Prac y todo su contenido de tu directorio personal de usuario.

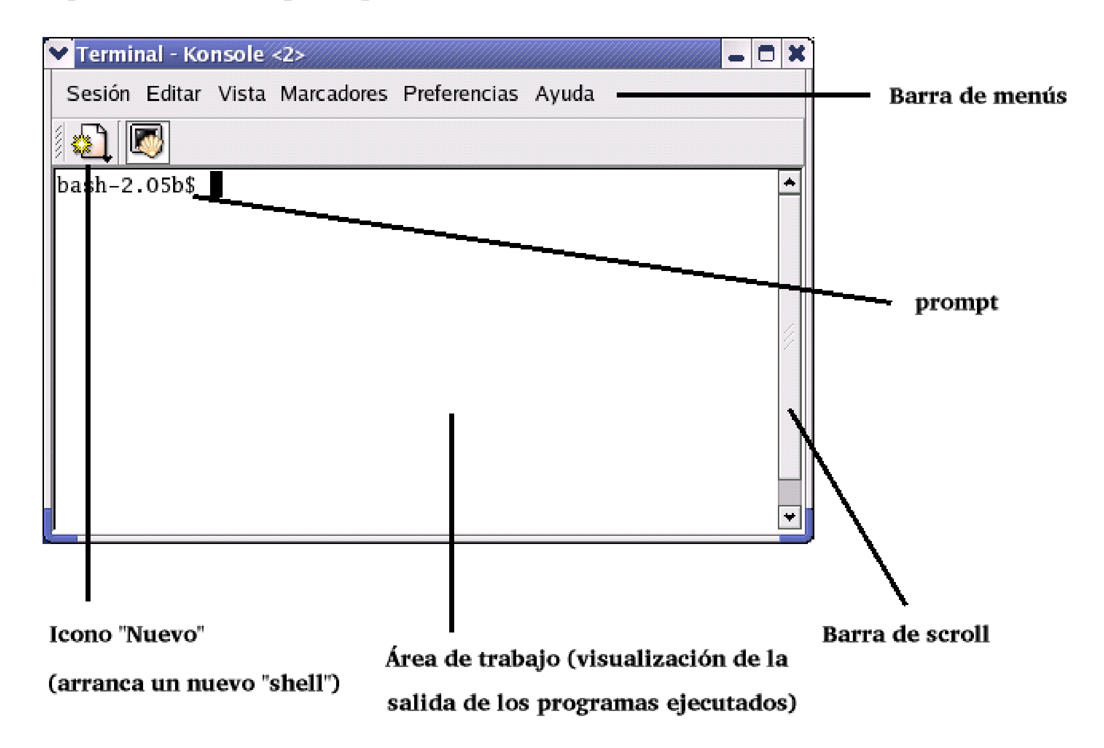
2.3. LA CONSOLA

Además del trabajo con entornos gráficos, GNU/Linux proporciona una herramienta muy potente para el trabajo cotidiano: la consola. Ésta no tiene una mera existencia anecdótica, como ocurría en Windows 98 con la ventana MS-DOS, sino que muchas veces (y más como informáticos que somos) recurriremos a su utilización. Al arrancar una consola se ejecuta un potente programa intérprete de órdenes (shell) en modo texto. Este intérprete no sólo permite ejecutar programas y órdenes básicas de manejo de ficheros, entre otras, sino que incluso proporciona un poderoso y eficaz lenguaje de programación específicamente destinado a la creación de scripts (macros) que permiten automatizar muchas de las tareas a realizar en nuestro ordenador.

Para arrancar una consola en el entorno de ventanas debemos desplegar el menú del **botón K** y elegir la opción **Terminal** dentro de **Herramientas del sistema**. Además, GNU/Linux arranca por defecto hasta seis consolas (modo texto) "fuera" del entorno gráfico (de hecho el entorno gráfico es como si fuese una consola más en GNU/Linux). Para nosotros, la única diferencia entre la terminal X y la consola del modo texto de GNU/Linux, radica en que desde la primera podemos ejecutar programas gráficos (p.e. xemacs), mientras que desde la segunda no. Trabajaremos con el intérprete de órdenes en la última práctica.

2.3.1. Manejo de la consola y consolas virtuales

La siguiente figura muestra las partes principales de la terminal X de KDE. Lo que aparece en la ventana justo después de la barra de menús es el prompt del intérprete de órdenes o shell. Esto indica que el intérprete está preparado para ejecutar nuestras órdenes (en forma de órdenes del sistema o ejecución de cualquier programa). En la parte derecha tenemos una barra que sirve para desplazarse verticalmente por la ventana y poder visualizar los contenidos de la misma. La capacidad de esta barra es limitada. Permite visualizar la salida de un programa relativamente larga sin necesidad de usar un paginador (p.e. less, como veremos en el Tema 5 de teoría). El icono Nuevo permite crear un shell independiente en la misma ventana (en la imagen, correspondiente al curso 04/05, este icomo aparece en la parte superior izquierda, en la aplicación actual lo verás en la parte inferior izquierda con un aspecto parecido).



Para acceder desde el modo gráfico a las consolas virtuales en modo texto de GNU/Linux habremos de pulsar la combinación de teclas Alt + Control + F? . Si F? es F1 accedemos a la consola 1, si es F2 a la consola 2, …, si es F6 a la consola 6. Una vez en modo texto, basta con teclear Alt + F? para cambiar entre consolas (ya sabes, F1 para la consola 1, F2 para la consola 2, …, F6 para la consola 6). Para

volver nuevamente al entorno gráfico de trabajo KDE tendremos que pulsar Alt + F7. Por defecto, GNU/Linux define estas seis consolas en modo texto. Dicho número se puede incrementar y en ese caso para volver al modo gráfico habría que pulsar Alt + F? siendo ? un número igual al número de consolas de texto que hemos definido más uno, pero no tiene realmente mucho sentido (y sobre todo si estamos usando el sistema X Window para trabajar en un entorno gráfico).

Ejercicio 20: Ejecuta un terminal X. Teclea df -h y pulsa Enter. Haz clic en el icono Nuevo. Teclea ls -l y pulsa Enter. Observa que aparece un nuevo icono en la parte inferior de la ventana. Desplázate de un shell a otro. Finalmente, teclea exit y pulsa Enter en el shell número 2. Cierra la ventana del terminal X.

2.4. MONTAJE Y DESMONTAJE DE UNIDADES

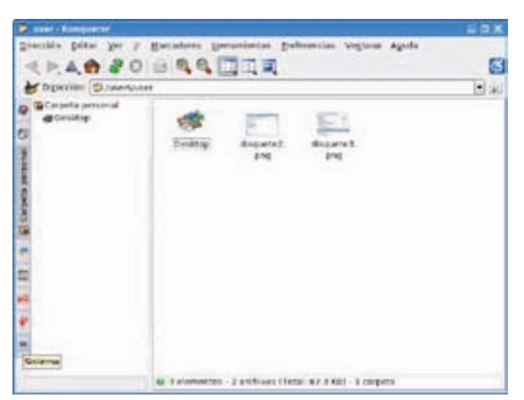
En GNU/Linux, como en el resto de sistemas operativos de la familia UNIX, existe un único sistema de ficheros a diferencia de Windows donde existe un sistema de ficheros por cada unidad (lo que pasa es que cuelgan artificialmente de Mi PC). Para poder acceder, pues, a los contenidos del sistema de ficheros de una unidad o dispositivo concreto (p.e. una memoria USB), hemos de montarlo en el sistema de ficheros de Linux. Una vez hayamos terminado de trabajar con dicha unidad o dispositivo, tendremos que desmontarlo y hasta que no se realice esta acción no debemos sacarlo físicamente del ordenador. Esto es muy importante, ya que si no procedemos de esta manera se pueden producir errores indeseables e incluso pérdida de datos.

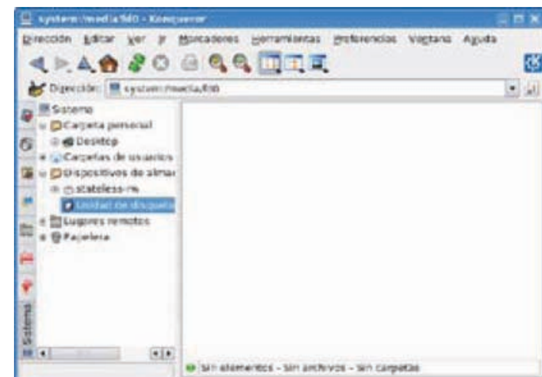
Tomemos como ejemplo el disquete. Quiero guardar una copia de un fichero en el disquete. Lo introduzco en la disquetera y lo monto. A partir de ese momento ya puedo acceder a los contenidos del disquete (lectura y escritura salvo que esté protegido). A continuación, procedo a realizar la copia del fichero deseado y, finalmente, desmonto la unidad (disquetera). En el momento en el que se apaga la luz de la disquetera, ya puedo extraer el disquete de la unidad.

Para montar un dispositivo desde una terminal en modo texto puedo usar la orden mount. Para desmontarlo se utiliza la orden umount. La sintaxis, en ambos casos, sería:

- *mount dispositivo*
- *umount dispositivo*

Sin embargo, el entorno KDE instalado en aulas dispone de herramientas que realizan el montaje de manera automática (en el caso de dispositivos USB). En otros casos, podemos usar Konqueror para realizar el montaje y desmontaje de unidades de forma cómoda. Por ejemplo, para montar el disquete arrancamos Konqueror (por ejemplo, abriendo la carpeta **Personal** que hay en el escritorio) y hacemos clic en el icono Sistema tal como indica la imagen de la izquierda en la siguiente figura

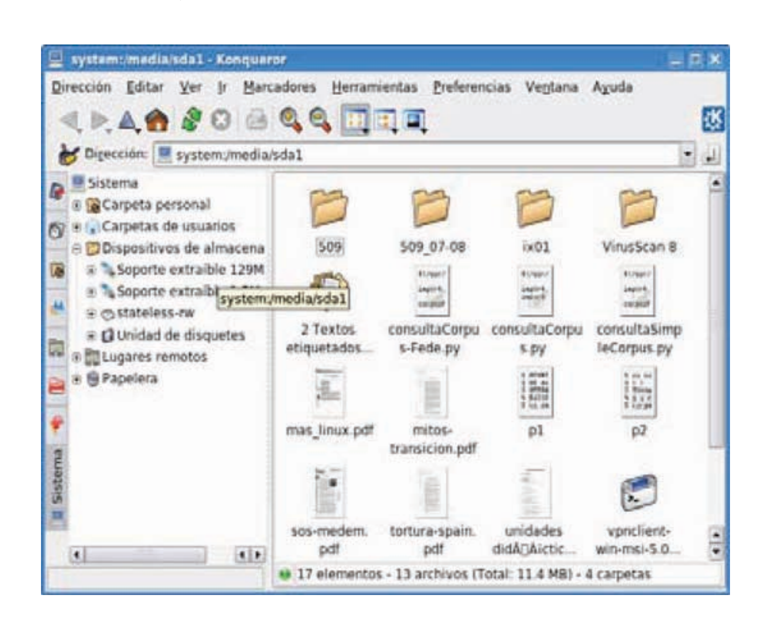




Entonces, el panel de la izquierda de Konqueror cambiará para mostrar los contenidos que ilustra la imagen de la derecha en dicha figura. Si desplegamos la carpeta Dispositivos de almacenamiento, tendremos acceso al icono Unidad de disquete. Su menú contextual (lo desplegamos haciendo clic con el botón derecho del ratón sobre el icono) contiene, entre otras, la opción **Montar**. Si la ejecutamos Konqueror mostrará en el panel de contenidos (el de la derecha) el contenido del disquete introducido en la disquetera. Si el disquete está montado, entonces el menú contextual mostrará la opción **Desmontar** en lugar de **Montar**. Ejecutándola, desmontaremos el disquete y podremos sacarlo tranquilamente una vez se haya apagado la luz de la disquetera.

Ahora bien, ¿qué hacer si quieres guardar datos en tu memoria (o lápiz) USB (también conocido como memstick USB)? Si lo introduces, observarás cómo parpadea su lucecita indicando que se activa. En unos segundos, aparecerá una ventana similar a la que muestra la imagen de la izquierda de la figura que se miuestra a continuación. Para tener acceso al dispositivo (montarlo), basta con hacer clic en el botón Aceptar de dicha ventana. Al hacerlo aparece una ventana de Konqueror que muestra los contenidos de la memoria USB insertada (veáse la imagen de la derecha de la siguientre figura: el dispositivo se monta, en este ejemplo, en el directorio /media/sda1).

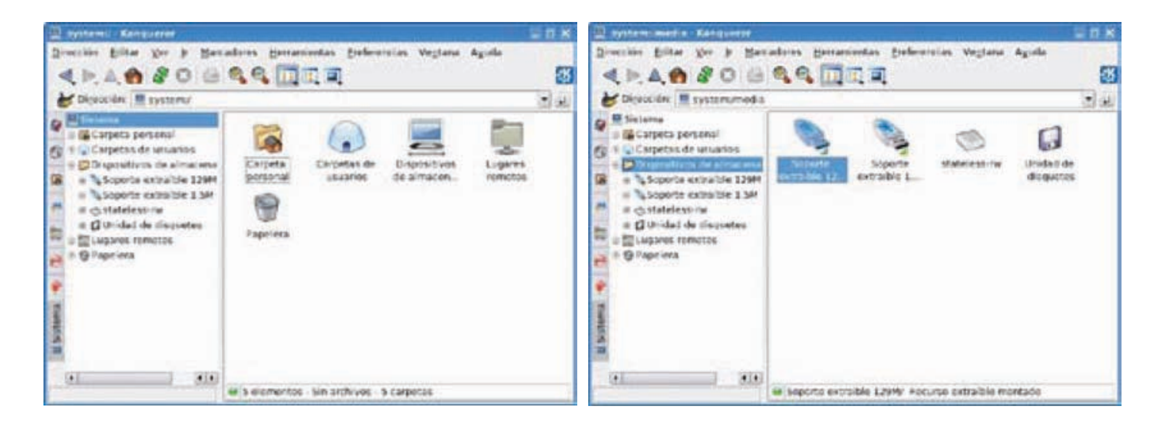




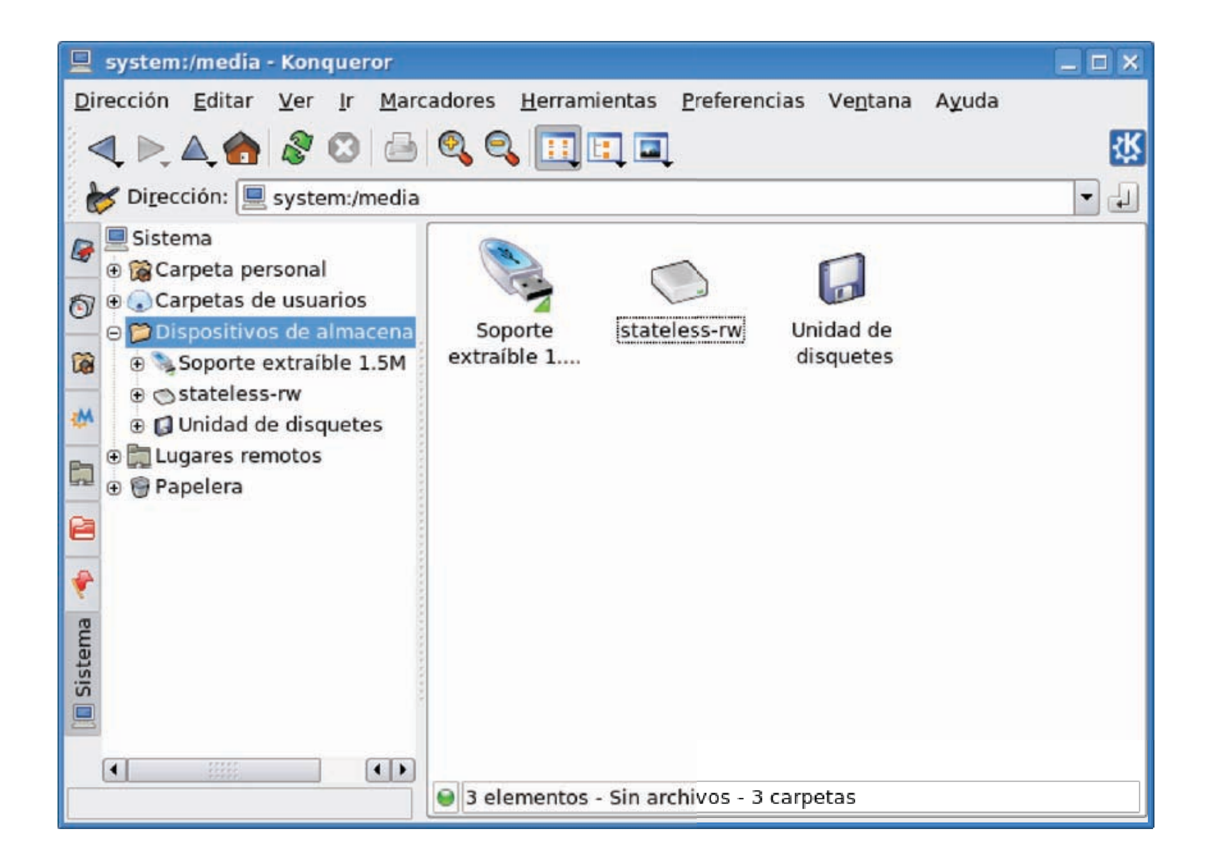
Es importante que tengas presente lo siguiente: muchos dispositivos USB de esta naturaleza contienen, en realidad, dos particiones. Es por ello que, cuando esto sea así, te aparecerán dos ventanas como las de la izquierda de la Figura ?. Si quieres, puedes montar las dos particiones (KDE las tratará como dos dispositivos USB independientes), aunque debes tener en cuenta que, normalmente, la primera de ellas ocupa muy poco espacio en el dispositivo y es usada por el fabricante para poner el manual en PDF y el driver para Windows. Nuestro consejo es que montes solamente la partición "grande" (al fin y al cabo es la que vas a usar). Para no montar la partición "pequeña" basta con hacer clic en el botón **Cancelar**.

Cuando hayas terminado de copiar tus datos en la memoria USB, tienes que desmontarla antes de poder extraerla del puerto USB donde está conectada al ordenador. La pregunta es, ¿cómo puedo hacerlo? Para ello, podrías seguir una estrategia parecida a la mostrada anteriormente con el disquete, pero en este caso localizando previamente la unidad USB que quieres desmontar. Sin embargo, nosotros vamos a enseñarte un método ligeramente diferente y que, además, te será muy útil para el manejo de dispositivos de almacenamiento y particiones en KDE.

En realidad, se puede tener un acceso centralizado a diversos dispositivos y carpetas de usuario mediante Konqueror. Para ello, basta simplemente con visitar la Dirección: system:/ (sólo una barra, tal como muestra la imagen de la izquierda de la figura que se muestra a continuación). Si hacemos click en el icono Dispositivos de almacenamiento, entonces accedemos a la dirección system:/media donde podemos observar, tal como indica la imagen de la derecha de siguiente figura, los iconos que representan al disquete y a las memorias USB (o particiones) que podamos haber montado, entre otros.



Para poder extraer la memoria USB una vez hemos terminado de usarla, hemos de acceder a este sitio y, empleando el menú contextual del icono que representa a nuestro memstick, ejecutar la opción Extracción segura. Una vez hayamos ejecutado esta opción y desaparezca el icono que representa a la memoria USB, tal como indica la siguiente figura, podemos extraer tranquilamente el memstick del puerto USB al que está conectado.



2.5. TERMINAR UNA SESIÓN DE TRABAJO

Para terminar una sesión de trabajo con GNU/Linux tendremos que cerrar, en primer lugar, la sesión X con KDE (entorno gráfico de usuario). Para ello:

- bien desplegamos el menú del **botón K** y elegimos la opción **Termina**r...

- bien hacemos clic con el botón derecho del ratón en cualquier superficie visible del escritorio para activar su menú contextual y elegimos la opción **Terminar <nombre>**.

En cualquiera de los dos casos, el escritorio de KDE se oscurecerá y aparecerá una ventana pidiendo confirmación de la acción realizada. Si hacemos clic en el botón **Terminar sesión actual**, entonces el proceso de desconexión continúa y, al salir de la sesión KDE, vuelve a aparecer la pantalla inicial de conexión donde se pide el nombre de usuario y la contraseña. Si hacemos clic en el botón **Cancelar**, la acción se aborta y podemos continuar trabajando en el entorno KDE como si nada

Ejercicio 21: Si llevas tu memoria USB, introdúcela en la conexión USB que hay en la parte frontal del ordenador de prácticas. Abre una ventana de Konqueror y accede a sus contenidos. Crea una carpeta en su interior y copia en ella un par de ficheros. Desmóntala y, luego, extráela mediante el mecanismo de extracción segura que hemos visto.

hubiese pasado. También podemos hacer clic en los botones **Apagar el ordenador** y **Reiniciar el ordenador**, que ejecutan las acciones pertinentes para apagar o reiniciar el equipo (obviamente).

Si estamos "situados" ante la pantalla inicial de conexión, tendremos que hacer clic en la opción **Apagar** para apagar definitivamente el ordenador. Al hacer clic en dicho botón, emerge una ventana que pide confirmación de esta acción. Si hacemos clic en **Apagar**, Linux apagará el ordenador tras realizar una serie de acciones. Si hacemos clic en **Cancelar**, Linux no hace nada. Antes de marcharnos, tendremos que apagar el monitor (éste no se apaga automáticamente).

CAPÍTULO 3

3. INTERNET: NAVEGADORES Y BUSCADORES

3.1. EL NAVEGADOR MOZILLA FIREFOX

World Wide Web (abreviadamente www o web) es un servicio que permite la publicación de información multimedia en Internet. Con esto queremos decir que es posible publicar documentos que incluyan texto, imágenes, sonido, películas, formularios interactivos y un largo etcétera en continuo crecimiento. Aunque es el servicio más conocido por la red y el que ha provocado su enorme expansión en los últimos años, no es el único, y por ello no hay que confundir *Internet* con la *web*.

Acceder a la información publicada mediante este servicio se suele denominar navegar y se realiza mediante programas especiales que se suelen denominar navegadores. *Mozilla Firefox* es uno de los múltiples navegadores que se pueden utilizar, y tiene la ventaja importante de que funciona igual sobre *Windows* que sobre *Linux*. Otros, como *Internet Explorer* (sobre Windows) o *Konqueror* y *Galeón* (sobre Linux), funcionan de modo muy similar, y es de esperar que habiendo aprendido a utilizar Firefox sea sencillo adaptarse a cualquiera de ellos.

3.1.1. Direcciones de páginas *web*

A la hora de acceder a una página *web* hay que conocer la dirección IP del servidor que la provee o, como es más habitual, su nombre DNS. Normalmente, las direcciones *web* tendrán una forma similar a la siguiente, que apunta a la página *web* de la UJI:

```
http://www.uji.es/
```

Esto es lo que se conoce como URL (*Uniform Resource Locator*). A continuación se van a describir cada uno de los elementos que componen esta URL:

- **http://** `http` son las siglas de *HyperText Transfer Protocol*, que es el protocolo utilizado por los clientes y servidores de Internet para transmitir y recibir información en forma de páginas *web*. Existen otros protocolos, menos utilizados en la actualidad, y que también son “reconocidos” por los navegadores, tal como `ftp`. Es necesario indicarle al navegador qué protocolo se va a utilizar, aunque la mayoría de navegadores, *Mozilla Firefox* entre ellos, incluyen por defecto y de forma automática `http://`.

- **www** www son las siglas de *World Wide Web* (la telaraña mundial) y suelen aparecer en la mayoría las direcciones *web* tras el protocolo, aunque esto dista mucho de ser una regla, dadas las numerosas excepciones. De hecho www es el nombre (o un alias) del servidor al que queremos acceder.

- **uji** este nombre, conocido como subdominio, indica, a grandes rasgos, dónde se encuentra el servidor al que queremos acceder dentro del nivel superior (que se explicará en el siguiente punto). En este caso, resulta claro que accedemos a la Universitat Jaume I. Ejemplos de otros nombres indicativos de dominios son `nature`, `la-moncloa`, `onu`, `greenpeace`, etc.

- **es** dominio de nivel superior, en este caso indicativo de país, que nos indica que la página *web* se encuentra en un servidor español. Otros indicativos de país serían `.fr` (Francia), `.uk` (Gran Bretaña), `.mx` (México), etc. Estos dominios de nivel superior también pueden ser, en lugar de indicativos de país, referencias al tipo de actividad desarrollado por el poseedor del servidor *web*, tal y como se indica en la siguiente tabla:

Dominio	Poseedor
.com	Empresas comerciales
.edu	Instituciones educativas americanas, desde escuelas primarias hasta universidades
.gov	Organizaciones del gobierno americano
.net	Organizaciones relacionadas con la comercialización o desarrollo de Internet
.int	Organizaciones internacionales
.mil	Organizaciones militares
.org	Organizaciones sin ánimo de lucro que no caben en las anteriores definiciones
.tv	Aunque en realidad es el dominio de un país, Tuvalu, antiguamente Islas Ellice, lo utilizan las cadenas de televisión

Existe una orden que permite averiguar el nombre y dominio de un ordenador a partir de su dirección IP y viceversa. En Linux, desde una consola, se puede ejecutar la orden `host`:

```
host direccion_IP
```

devuelve el nombre y dominio del ordenador que tiene asociada la dirección indicada.

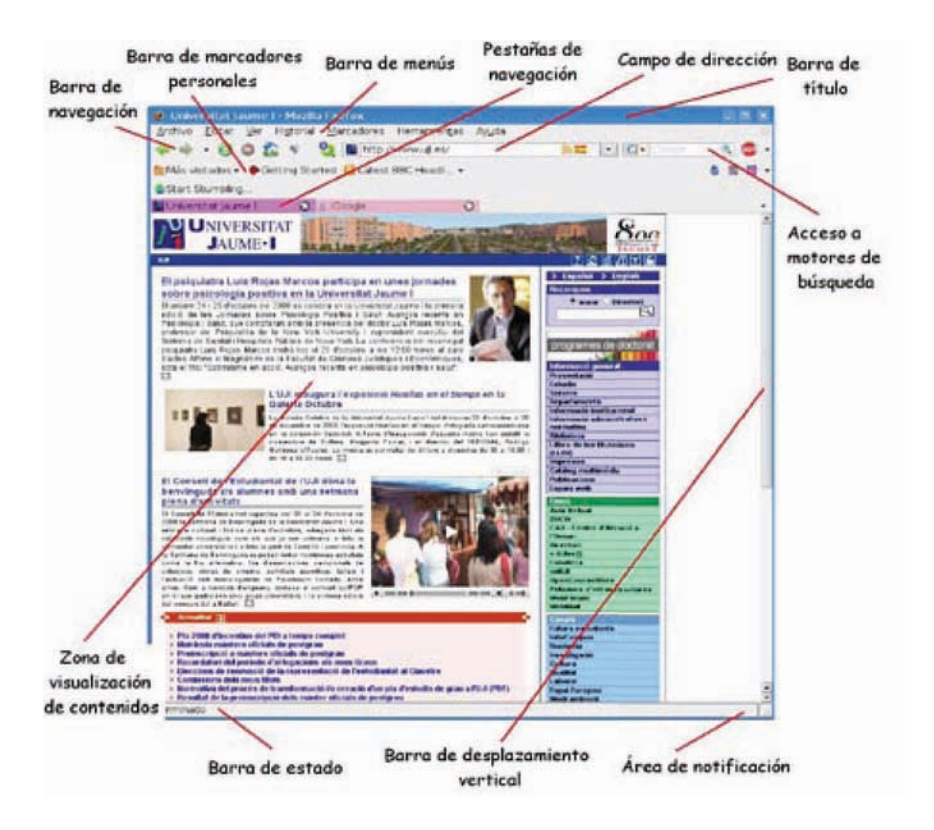
```
host nombre.subdominio.dominio
```

devuelve la dirección IP del ordenador que tiene asociado el nombre indicado.

3.1.2. Componentes de la ventana de *Mozilla Firefox*

Para arrancar la aplicación, si estamos trabajando con el sistema operativo Linux y bajo el entorno KDE, podemos pulsar sobre el **botón K**, **Internet**, **Firefox Web Browser**.

Tras arrancar la aplicación, aparecerá la ventana de *Mozilla Firefox* con un aspecto similar al que muestra la siguiente figura. Veamos de modo resumido cuál es la función de cada uno de los componentes que podemos ver en dicha figura.



■ La *barra de título* de la ventana contiene el título que su autor ha dado a la página mostrada.

■ La *barra de menús* agrupa las distintas órdenes de manejo de la aplicación.

■ La *barra de navegación* contiene una serie de botones que permiten ejecutar de modo rápido las funciones de *Mozilla Firefox* que se emplean con mayor frecuencia.

■ El *campo de dirección* contiene la dirección de la página mostrada.

■ La *barra de marcadores personales* permite acceder a enlaces propuestos por *Mozilla Firefox* o añadidos por el usuario, sin tener que recordar su dirección.

■ La *zona de visualización de contenidos* contiene la página actual. Si su contenido es mayor del que cabe en la ventana, aparecen unas barras de desplazamiento horizontal y vertical que nos permiten desplazarnos por el documento.

■ La *barra de estado* (esquina inferior izquierda) puede contener texto describiendo la dirección de la página apuntada, algún mensaje de error u otras informaciones.

■ El *área de notificación* (esquina inferior derecha) puede mostrar indicadores para una serie de eventos. Lo más frecuente es que aparezca el indicador de seguridad (icono con forma de candado): éste indica si la página mostrada se ha transferido mediante un método seguro o no y permite acceder a las opciones de

seguridad del programa. Solamente aparece este indicador si hemos accedido a una dirección del tipo `https` (por ejemplo, `https://e-ujier.uji.es`).

- *Firefox* permite navegar con pestañas, tal como veremos en la sección 1.4. Las pestañas permiten mostrar el contenido de diversas páginas web agrupadas en una única ventana del navegador, lo cual resulta muy cómodo en muchas ocasiones. Podemos ver los indicadores de cada pestaña abierta (área marcada como *pestañas de navegación* en la figura anterior) debajo de la barra de marcadores personales.

- En la esquina superior derecha tenemos el *acceso a diversos motores de búsqueda*, como por ejemplo el de *Google*. Podemos realizar búsquedas directas simplemente tecleando las palabras clave en dicho cuadro. También tenemos acceso a un cuadro desplegable con la lista de motores de búsqueda instalados en nuestro navegador.

Para salir de *Mozilla Firefox* se usa la orden **Archivo –> Salir**. No debemos confundir esta orden con **Archivo –> Cerrar**, que cierra la ventana activa en el caso de tener varias ventanas abiertas con *Mozilla Firefox*. Si sólo tenemos una ventana abierta, **Archivo–> Cerrar** la cierra, saliendo a la vez de *Mozilla Firefox*.

3.1.3. Primeros pasos de navegación con *Mozilla Firefox*

Cuando accedemos a una página, estamos visualizando información en pantalla que puede encontrarse en uno o en varios servidores distintos de la red. Las páginas están escritas utilizando un lenguaje especial de etiquetas denominado HTML, que permite integrar textos, imágenes, sonidos y toda una serie de documentos de distintos tipos.

Al utilizar este formato, el texto pasa a denominarse *hipertexto*. Básicamente esto quiere decir que partes del texto se comportan como enlaces a otros documentos. Los enlaces aparecen normalmente como texto subrayado y en otro color (en general azul), o con otras características que los hacen fácilmente identificables. Además, si ponemos el cursor del ratón sobre ellos, éste cambia su forma a una mano o algo similar. Si pulsamos sobre uno de ellos, visualizaremos en pantalla la página asociada.

El hecho de visualizar distintas páginas y de acceder desde unas a otras se denomina, como ya se ha mencionado, navegar. Existen muchas formas de navegar y de acceder a la información contenida en *Internet*. De modo resumido, podemos acceder a una página *web* usando los siguientes métodos:

- Escribiendo la dirección de la página en el campo de dirección del navegador y pulsando la tecla Enter.

- Pulsando un enlace en una página visualizada con el navegador.

- Cuando mantenemos el cursor sobre un enlace, este se convierte en una pequeña mano, y en el campo de mensaje de estado se muestra la dirección a la que apunta el enlace o bien un comentario al respecto introducido por el creador de la página. Pulsando el botón derecho del ratón sobre un enlace se despliega un menú de contexto, una de cuyas opciones es **Abrir enlace en una ventana**

nueva. De esta forma, abrimos la dirección indicada en el enlace en una ventana distinta a la original. Otra opción es **Abrir enlace en una pestaña nueva**, que crea una barra de pestañas si no existe y abre el enlace en una nueva pestaña de la barra. Lo puedes probar con el enlace a la página del `Llibre Electrònic d'Universitat (LLEU)` de la UJI que tienes en `http://www.uji.es/`.

■ Usando una dirección guardada previamente en el menú **Marcadores**.

■ Utilizando los botones **Anterior** y **Siguiente**. El botón **Anterior** permitirá visualizar de nuevo la página anterior si hemos accedido a la actual desde uno de sus enlaces. El botón **Siguiente** permitirá volver a visualizar una página si hemos regresado de la misma. Ambos botones pueden ser accedidos de tres modos, aunque aparezcan con nombres distintos en cada caso:

◆ Mediante la barra de navegación.

◆ A través del menú **Historial**.

◆ Mediante el menú contextual desplegable con el botón derecho del ratón pulsado sobre los contenidos de la página actual (**Anterior/Siguiente/Recargar**).

■ Utilizando las direcciones que hemos ido recorriendo en la sesión actual de navegación y que se muestran en el menú **Historial**.

Otras dos acciones básicas que se suelen utilizar a la hora de navegar son:

■ **Recargar** la página actual mediante el botón correspondiente de la barra de navegación.

■ **Parar** la carga de la página actual mediante el botón correspondiente de la barra de navegación.

3.1.4. Navegación con pestañas

La navegación con pestañas proporciona una forma cómoda de navegar por la red. *Firefox* permite navegar entre múltiples sitios web dentro de una sola ventana usando pestañas de navegación.

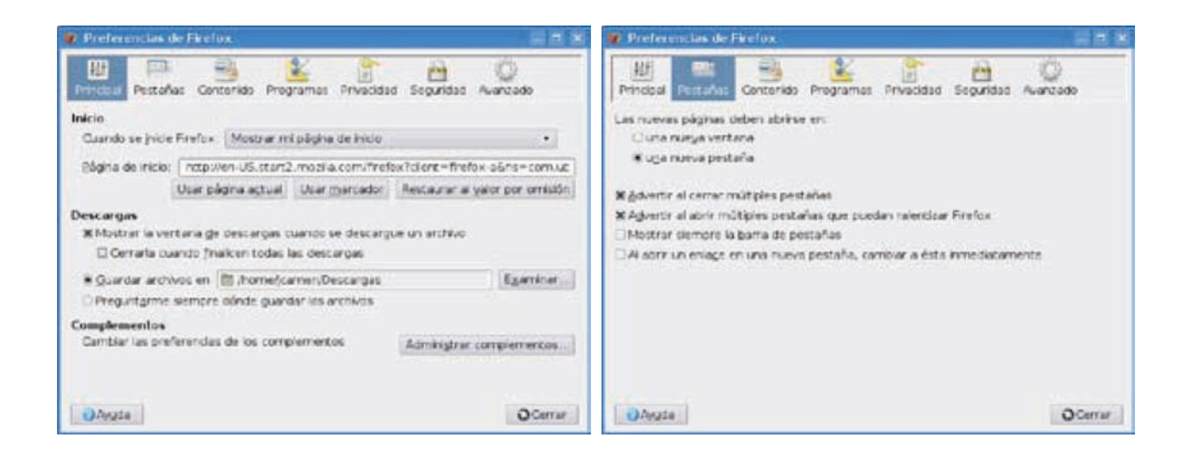
Mediante la combinación de teclas Ctrl + T podemos abrir una nueva pestaña en blanco. Otra forma de abrir una pestaña es haciendo *clic* con el botón derecho del ratón sobre un enlace y seleccionar la opción **Abrir enlace en una pestaña nueva** del menú contextual que aparece. Entonces, podemos continuar la navegación en la nueva pestaña.

Una vez tengamos varias pestañas abiertas podemos circular entre ellas mediante la combinación de teclas Ctrl + v Pág y Ctrl + Re Pág.

3.1.5. Configuración del navegador

La opción **Preferencias** del menú **Editar** presenta una ventana con un conjunto de opciones agrupadas que permiten configurar determinados aspectos del navegador (véase la figura que se presenta a continuación). Al seleccionar esta opción aparecen una serie de pestañas con las partes que se pueden configurar. Algunas de las

opciones que se pueden configurar se detallan a continuación:



En la pestaña *Principal* (véase la figura anterior izquierda) podemos, entre otras cosas, fijar las opciones de:

- **Página de inicio**. En este cuadro podemos especificar la página principal que se cargará en el navegador cuando se inicia la ejecución de la aplicación, o en cualquier momento que se pulse el botón de **Inicio**. Para seleccionar la página de inicio hay que hacer *clic* en la pestaña *Principal*. Debajo del cuadro donde se especifica la página de inicio aparecen las diferentes opciones que se tienen para seleccionar la página de inicio del navegador:

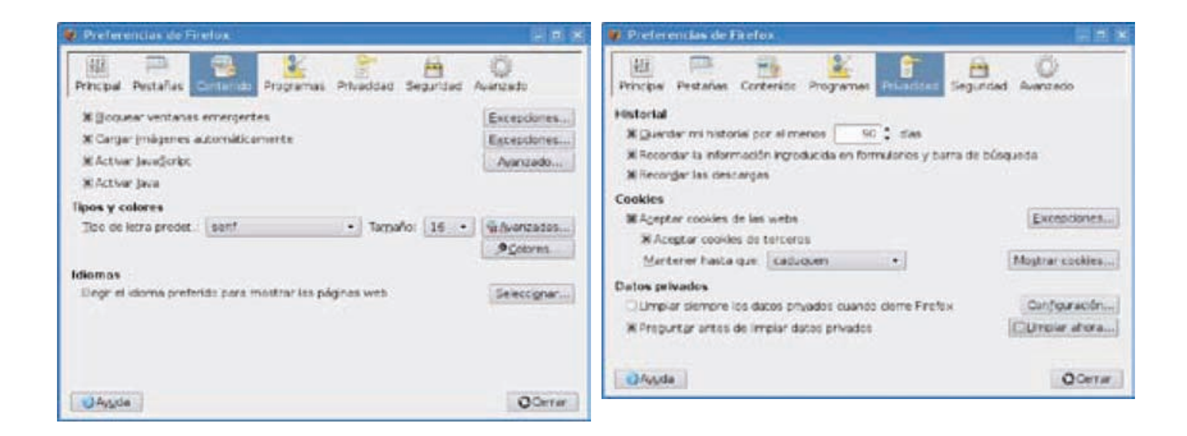
    - **Usar página actual:** configura como página de inicio la página que está abierta en este momento en el navegador.

    - **Usar marcador:** permite elegir un marcador para que sea la página de inicio.

    - **Restaurar al valor por omisión:** sirve para colocar nuevamente la página de inicio con la que, por defecto, viene configurado el navegador.

- **Guardar archivo en** o **Preguntarme siempre dónde guardar los archivos**. Aquí podemos especificar una carpeta por defecto en la que se guardarán todos los archivos que descarguemos, o bien podemos indicarle a *Firefox* que muestre un cuadro de diálogo para seleccionar la carpeta donde vamos a guardar cada fichero que descarguemos.

En la pestaña *Pestañas* (valga la redundancia) encontramos opciones para configurar los aspectos relativos al manejo de las pestañas de navegación (véase la figura anterior derecha). Por ejemplo, podemos indicar si deseamos que cada nueva página que se abra al hacer *clic* en un enlace lo haga, bien en una nueva ventana del navegador, o bien en una nueva pestaña en la *misma* ventana de navegación. También podemos fijar el nivel de “alerta” (mensajes de aviso que mostrará el navegador) por si decidimos cerrar una ventana con múltiples pestañas abiertas o bien estamos abriendo demasiadas pestañas (con lo que aumenta el consumo de memoria RAM).



En la pestaña *Contenido* (véase la figura anterior izquierda) accedemos a las opciones que permiten:

- **Bloquear ventanas emergentes**. Al activar esta casilla, *Firefox* impide la aparición de nuevas ventanas lanzadas por una página a la que estemos accediendo.

- **Cargar imágenes automáticamente**. Cuando la desmarcamos, se desactiva la carga de imágenes en las páginas web a las que accedemos. Tan sólo leeremos el texto, y aparecerán "huecos" en los lugares donde deberían aparecer las imágenes. Si tenemos una conexión lenta a Internet puede resultar conveniente tener desactivada esta opción, de forma que cuando nos conectemos a una dirección dada sólo se transferirá el texto de la página visitada.

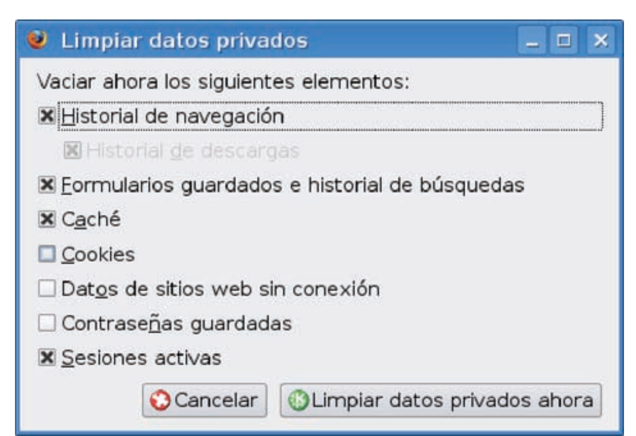
Para ambas opciones podemos definir **Excepciones**. Haciendo *clic* en el botón correspondiente, podemos especificar las direcciones de aquellas páginas web en las cuales no se va a aplicar la normativa definida en cuanto a ventanas emergentes y carga de imágenes.

En la pestaña *Privacidad* (véase la figura anterior derecha) tenemos las secciones que permiten manejar:

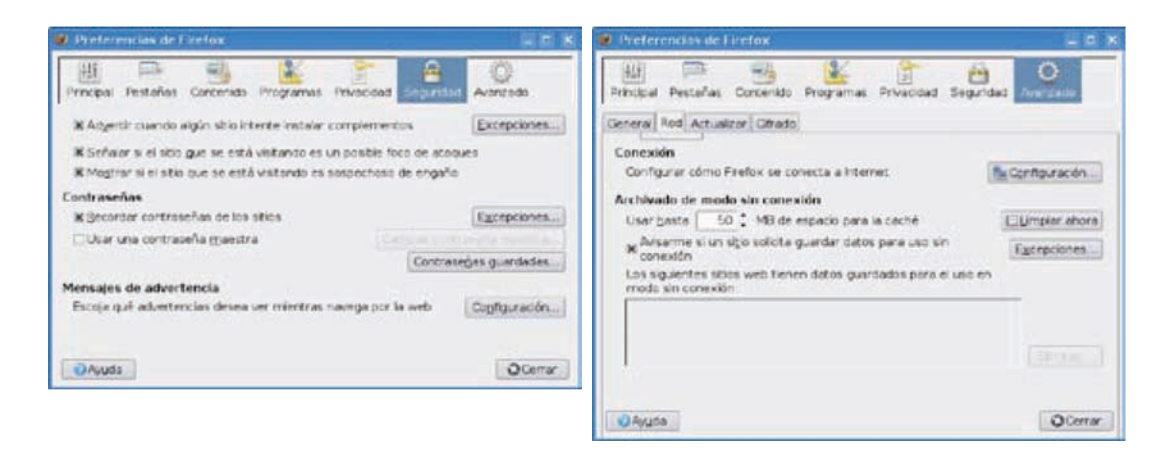
- El **Historial** de navegación. Esto hace referencia a los datos que *Firefox* guarda durante las diversas sesiones de navegación (imágenes en caché, direcciones de páginas web visitadas, datos introducidos en formularios,…).

- Las **Cookies**. Las *cookies* son datos que los servidores que visitamos envían a nuestro navegador para que los almacene de alguna manera (en disco, por ejemplo). Estos datos son necesarios para dar cuenta de la sesión de navegación con estos servidores, y así poder gestionar por ejemplo, cosas como una cesta de la compra (compras por Internet) o la identificación en un portal de servicios (como el de la UJI). En esta sección podemos definir las opciones de nuestra "política" al respecto de la aceptación o rechazo de las *cookies* y de su caducidad.

■ Los **Datos privados**. Si hacemos *clic* en el botón **Limpiar ahora...** se visualiza la ventana que vemos en la figura de la derecha. Cabe destacar las opciones que permiten vaciar la **Caché**, que es el espacio de disco ocupado por la información que se descarga el navegador de los servidores (imágenes, por ejemplo) y que permite acelerar la carga de las páginas en posteriores visitas, las **Cookies** y los datos guardados en **Formularios** y el **historial de búsquedas**. A esta opción también se puede acceder a través del menú **Herramientas**, opción **Limpiar datos privados...**.



La pestaña *Seguridad* permite activar o desactivar diversas opciones de seguridad. Se recomienda no tocar las opciones que vienen por defecto a no ser que se sepa muy bien lo que se está haciendo. La siguiente figura muestra las diversas opciones que podemos configurar.



La pestaña *Avanzado* agrupa contenidos de diversa índole. Precisamente lo hace empleando igualmente *pestañas*. Nosotros nos centraremos en la pestaña *Red*, cuyos contenidos podemos apreciar en la imagen superior derecha.

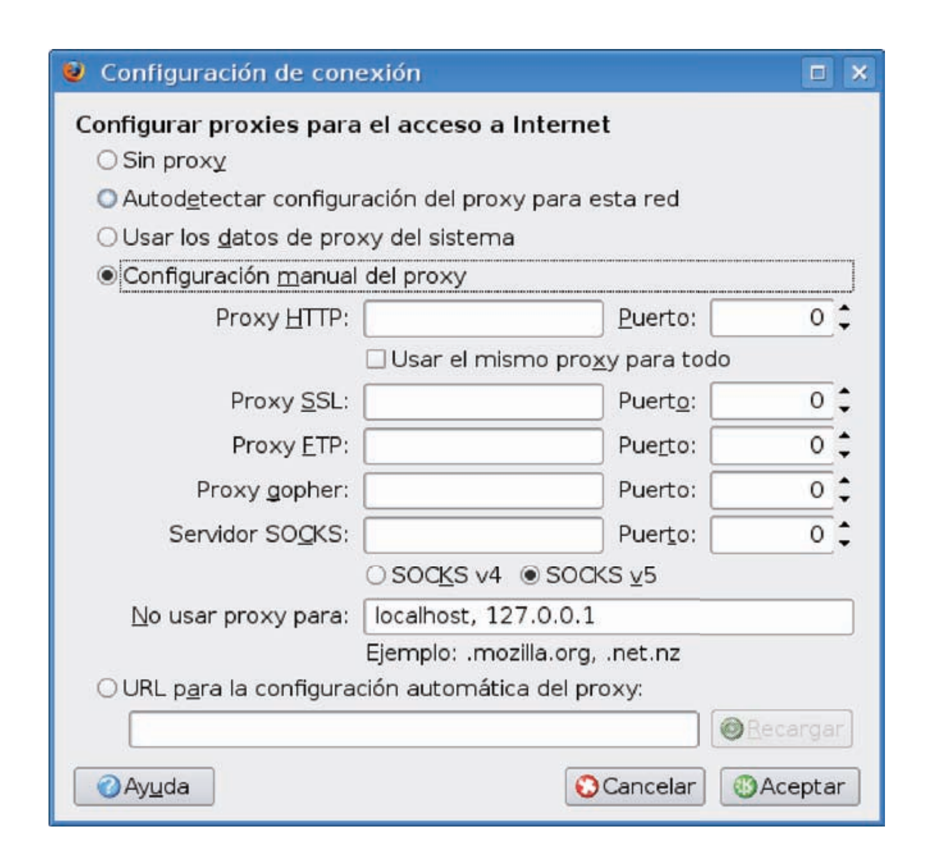
En la sección **Archivado de modo sin conexión** podemos definir el tamaño de la caché de disco que va a usar *Firefox*. La caché del navegador es el espacio de memoria y de disco destinado a almacenar copias de las últimas páginas visitadas. Si hacemos *clic* en el botón **Limpiar ahora** podemos borrar los ficheros almacenados hasta el momento. El navegador, a la hora de buscar una página, comprueba si ésta está en la caché, si la encuentra aquí, la búsqueda es mucho más rápida que si tiene que buscarla en Internet, puesto que únicamente tiene que leerla del disco o de la memoria del equipo local que está ejecutando el navegador. Se puede dejar como tamaño para la caché el que pone por defecto. Debes tener en cuenta, que mientras

navegas se almacena información de los sitios que has visitado. Desde esta pantalla se puede limpiar toda la información (botón **Limpiar ahora**).

En la sección **Conexión** se puede configurar la manera en la que *Firefox* se va a conectar a Internet. Básicamente nos va a permitir especificar un *proxy*, que es un servicio que permite gestionar una especie de memoria caché global compartida por todos los usuarios de una red local. En ella se almacenan las últimas páginas visitadas por todos los usuarios de la red local. Cuando un usuario busca una nueva página en Internet, el navegador comprueba si esta página está almacenada en la caché local del equipo que hace la búsqueda. Si no la encuentra, la búsqueda continúa en esta caché compartida (si está configurada). Sólo en caso de no encontrarla se realiza la búsqueda en Internet. Esta forma de trabajar reduce considerablemente el tiempo medio de búsqueda de las páginas solicitadas. Debes tener en cuenta que la utilización del *proxy* puede mostrar la información de la página sin las últimas actualizaciones que se han realizado. Puedes forzar la carga de la página desde Internet, usando el botón **Recargar** de la *barra de navegación*.

Para configurar el tipo de conexión de que disponemos hay que hacer *clic* en el botón **Configuración...**. Entonces, se mostrará la ventana que vemos en la Figura ?. La opción que se debe tener seleccionada para beneficiarse de esta caché común en la UJI es **URL para la configuración automática del proxy** y la dirección que se deberá escribir en el cuadro de abajo (URL) es:

- `http://www.uji.es/proxy.pac`,
- o bien `http://www.uji.es/sinpublicidad.pac`, si queremos eliminar los *banners* publicitarios ("pancartas" con publicidad).

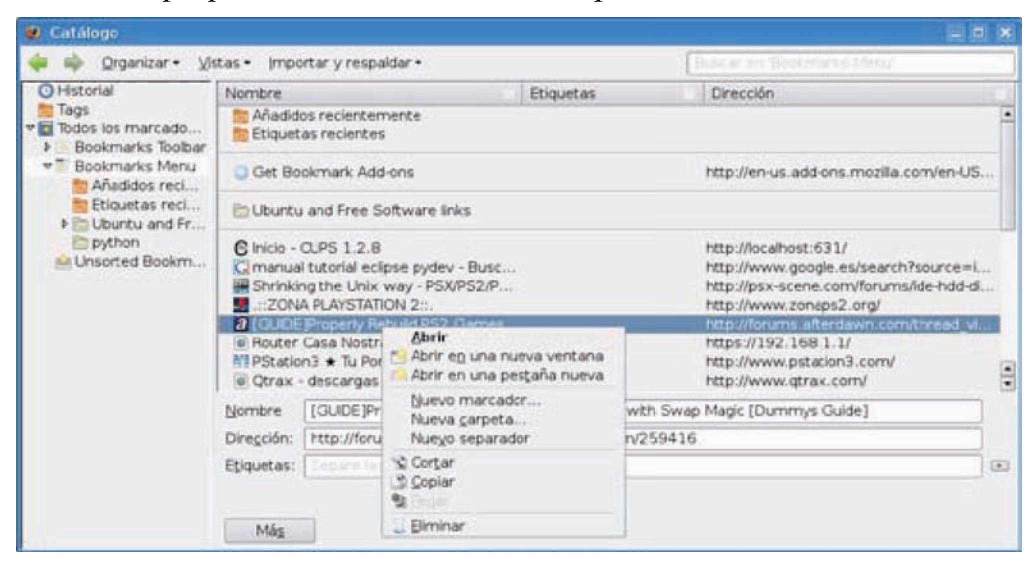


3.1.6. Creación y gestión de marcadores

El navegador *Mozilla Firefox* dispone de un fichero HTML donde se apuntan las direcciones *web* de páginas que se desean recordar. El navegador dispone de un conjunto de direcciones de páginas anotadas por defecto. Este fichero está organizado en carpetas donde se almacenan las direcciones de las páginas según temas. Esta lista se puede modificar añadiendo o eliminando direcciones de la misma. Para actuar sobre los marcadores, hay que usar el menú **Marcadores**, lo que permitirá elegir entre:

- **Añadir esta página a marcadores:** crea un marcador para la página actual, pudiendo seleccionar su ubicación.

- **Organizar marcadores...:** permite realizar todo tipo de operaciones con los marcadores, como crear nuevas carpetas, mover marcadores entre carpetas, borrar marcadores, etc. La Figura que se muestra a continuación muestra el aspecto que tiene la ventana que aparece al seleccionar esta opción. En ella, se ha hecho *clic* con el botón derecho del ratón sobre un marcador y podemos apreciar el menú contextual que permite realizar una serie de operaciones con él.

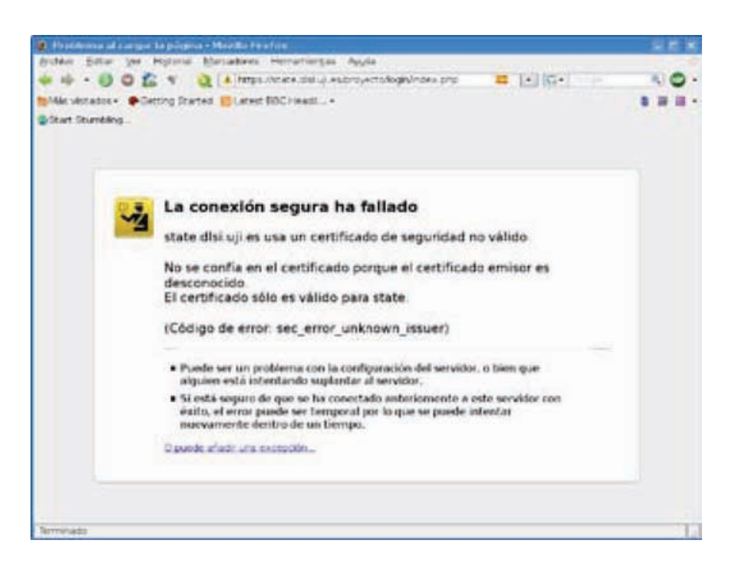


En la ventana para la organización de los marcadores tenemos herramientas para crear carpetas –que permiten agrupar marcadores sobre una misma temática, por ejemplo– para poner separadores –líneas de separación que permiten visualizar mejor los diferentes grupos de marcadores que hemos dispuesto– y para copiar, mover de sitio y eliminar marcadores. Podemos ejecutar estas opciones, bien a través del menú contextual que aparece al hacer *clic* con el botón derecho del ratón sobre un marcador, carpeta o separador, bien mediante las opciones que presenta el menú **Organizar**.

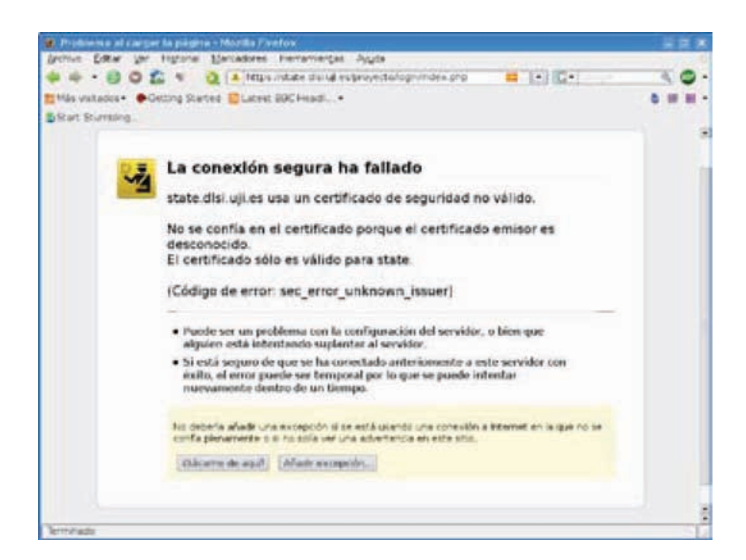
Para visitar un marcador previamente almacenado, basta con acceder a **Marcadores**, a continuación a la carpeta correspondiente y, por último, seleccionar el marcador deseado.

3.1.7. *Firefox 3.x.x* y los certificados

A partir de la versión 3.0 de *Firefox*, ha cambiado la forma en la que este navegador realiza el tratamiento de aquellos certificados considerados como "sospechosos" (de procedencia dudosa). Cuando navegamos en modo seguro –recuerda, se usa el protocolo `https` en lugar de `http` en el *campo de dirección* del navegador y aparece el candado cerrado en el *área de notificación*– es necesario que una entidad certificadora (un organismo reconocido a nivel internacional, como por ejemplo VeriSign) garantice la identidad del sitio al que nos estamos conectando y al que se supone que vamos a suministrar información confidencial.



En versiones anteriores del navegador *Firefox*, cuando la entidad certificadora empleada no era conocida por el navegador, éste se limitaba a mostrar un simple aviso y, tras aceptarlo, el usuario podía continuar tranquilamente con la conexión. Sin embargo, desde la versión `3.0` de *Firefox*, este comportamiento ha cambiado: el navegador es mucho más estricto en sus políticas de seguridad y va a requerir, en estas situaciones (entidad certificadora desconocida), que creemos una *excepción de seguridad* (véase la figura anterior).



Este hecho resulta relevante ya que los certificados que emplea la UJI para la identificación en su portal y acceso a pasarela Webmail y e-ujier no han sido expedidos por una autoridad certificadora conocida por el navegador. Si estás accediendo desde un ordenador de la UJI (como los del laboratorio o el aula de acceso libre), no tendrás problemas ya que se han creado las correspondientes excepciones de seguridad. Sin embargo, sí que puedes tenerlos si accedes desde el ordenador de tu casa o bien desde tu portátil conectado a la red Wi-Fi de la UJI.

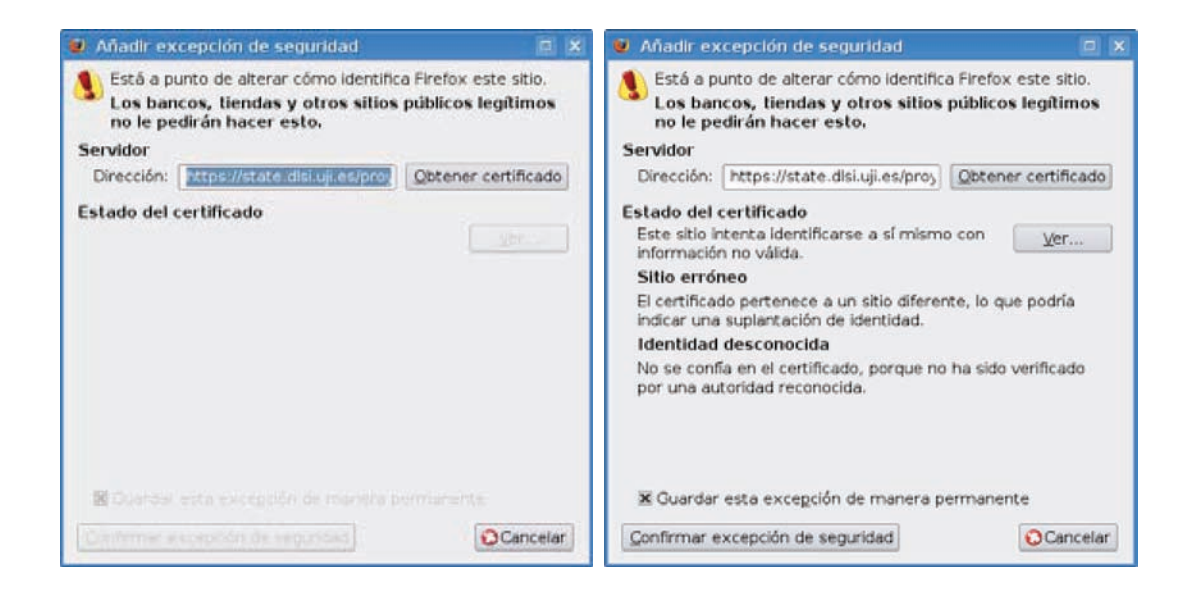
Es por ello, que vamos a mostrar en esta sección cómo crear una excepción de seguridad. Pensarás que esto no es necesario, que basta con usar otro navegador (*Internet Explorer*, por ejemplo). Bueno, pero ten en cuenta que las cuestiones de seguridad en la navegación por Internet no son para tomárselas a broma y *Firefox es el navegador más seguro en la actualidad*. *Internet Explorer* sigue esta línea: su próxima versión tendrá un comportamiento muy similar al que tiene *Firefox* ahora, por lo que resulta conveniente que vayas familiarizándote con estas prácticas.

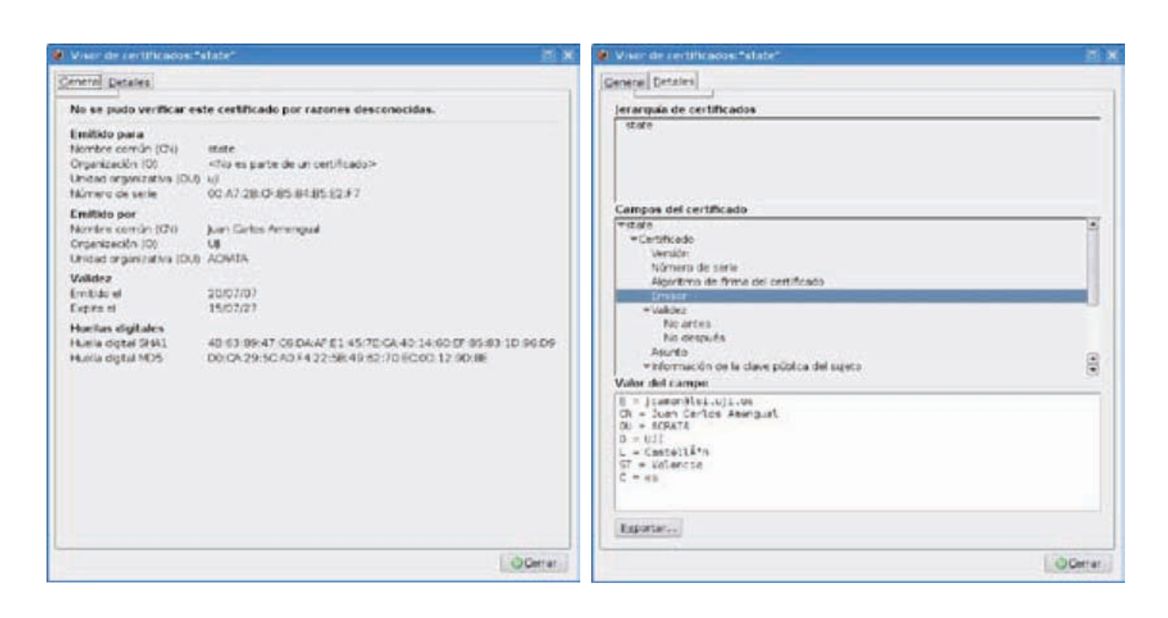
1. Nos conectamos a un lugar seguro (observa el empleo de `https` en el *campo de dirección*), pero el certificado ha sido emitido por una autoridad que *Firefox no reconoce*: Aunque esa pantalla puede inducir a pensar que hemos cometido un error, nada más lejos de la realidad: tan sólo nos está avisando del riesgo que podemos estar cometiendo si nos conectamos a este sitio. Observa que el navegador nos ofrece una posible solución: haz *clic* en **O puede añadir una excepción...**

2. Entonces, nos encontramos con la ventana que muestra la figura anterior. Tenemos dos posibilidades: salir o continuar añadiendo la excepción. Si haces *clic* en **¡Sácame de aquí!** , se aborta el proceso de conexión, mientras que haciendo *clic* en **Añadir excepción...** nos disponemos a añadir definitivamente una excepción de seguridad.

3. A continuación, en la ventana emergente, haremos *clic* en el botón **Obtener certificado** para así obtener los datos del certificado y poder observarlos detenidamente; véase la parte izquierda de la siguiente figura. Tras pulsar dicho botón la pantalla muestra la imagen de la derecha de dicha figura. Haremos *clic* en el botón **Ver...** para inspeccionar los detalles del certificado.

4. Si hemos pulsado el botón **Ver...** en el paso anterior, podemos acceder a los contenidos del certificado en una nueva ventana (véase la imagen derecha de la anterior figura). En la pestaña *General*, nos fijamos en quién ha emitido el certificado y para quién lo ha emitido. Si hacemos *clic* en la pestaña *Detalles* (ventana de la derecha), podemos obtener más información acerca del emisor del certificado. Si el emisor es conocido por nosotros y resulta de fiar, podemos proceder tranquilamente a cerrar esta ventana y hacer *clic* en el botón **Confirmar excepción de seguridad** de la ventana que aparece a la derecha en la siguiente figura. En ese momento, el certificado será aceptado y el navegador establecerá la conexión solicitada como si nada hubiese pasado.





3.2. BUSCADORES

La enorme expansión del servicio *web* en Internet hace que lo difícil no sea tanto encontrar información como saber ser selectivo al buscarla. Cada día es más importante la información sobre cómo se organiza y busca otra información. Podemos clasificar las distintas herramientas de búsqueda de páginas *web* en base a diversos criterios. En función de cómo se almacena la información en el buscador y cómo se presentan los resultados al usuario, podemos diferenciar entre *motores de búsqueda* y *catálogos*.

En los primeros, las páginas se almacenan a un mismo nivel y se presentan al usuario como una lista uniforme. En el caso de los *catálogos* las páginas se clasifican en un árbol de categorías en función de su contenido y los resultados se presentan diferenciando las categorías a las que pertenecen las distintas páginas

encontradas. Pasamos a continuación a explicar someramente el funcionamiento de dos de los motores de búsqueda más utilizados.

3.2.1. Motores de búsqueda

El tipo de búsqueda efectuado más habitualmente por los internautas es, sin duda, la búsqueda mediante la introducción de palabras clave. En este sentido, existen cada vez más herramientas que nos permiten utilizar este método. Su principio es simple: el buscador crea previamente un índice con un cierto número de páginas *web* de todo el planeta o restringido a alguna zona, tema o idioma concreto. La creación de este índice se suele efectuar por el texto completo, y todas las palabras de todas las páginas a las que se hace referencia se convierten en entradas del índice, para ser potenciales objetos de búsqueda.

El usuario introduce en el formulario del buscador una o más palabras clave, que pueden estar enlazadas entre sí mediante símbolos especiales (+ y –, por ejemplo), y el motor busca las ocurrencias de estas palabras en las páginas fuente almacenadas. Se establece, a continuación, una clasificación de las páginas más relevantes sobre el tema buscado y se presenta finalmente una lista de enlaces hacia dichas páginas.

Existe un gran número de buscadores por palabras clave: dos de los más conocidos son, sin duda, *Altavista* (`http://www.altavista.com`, o su versión en castellano denominada *Altavista España* `http://es.altavista.com`), y *Google* (`http://www.google.com`).

3.2.1.1. Altavista



Si accedemos al buscador *Altavista* a través de la URL `http://es.altavista.com`, podremos ver la pantalla inicial del buscador, que es similar a la captura que muestra la figura anterior. En la parte superior de la ventana podemos ver que el buscador contiene diversos paneles que nos permiten realizar distintos tipos de búsquedas. Así, el primero, **Web**, permite realizar búsquedas mediante palabras clave utilizando una sintaxis más o menos compleja. Los tres siguientes, **Imágenes**, **MP3/Audio** y **Vídeo**, permiten buscar distintos tipos de documentos gráficos y de sonido.

Con la búsqueda **principal** o **simple**, que se utiliza por defecto, podremos realizar la mayoría de nuestras peticiones. Sin embargo, si queremos establecer requisitos más complejos que deban cumplir las páginas encontradas, podemos utilizar ciertos

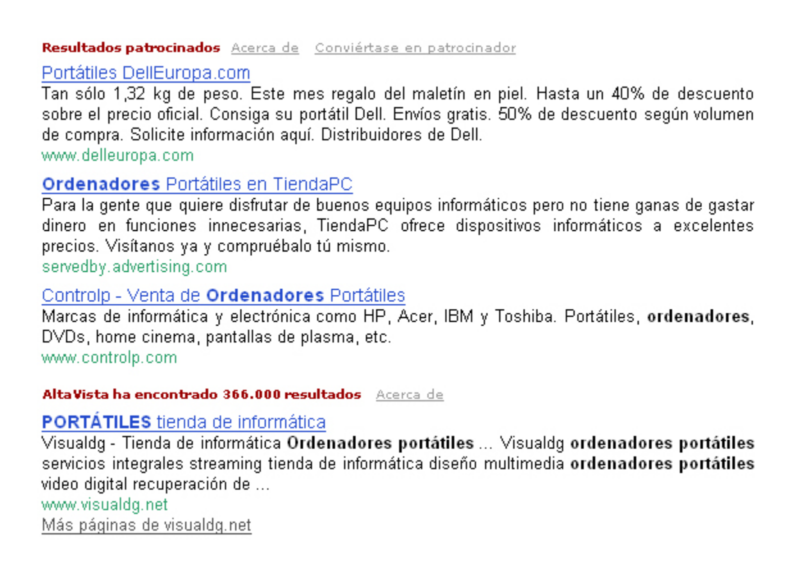
mecanismos de selección adicionales accediendo al modo de **Búsqueda avanzada**.

En el recuadro del formulario escribiremos el criterio de búsqueda, que podrá ser una o varias palabras con algunos símbolos especiales que nos permiten restringir los resultados encontrados. Una vez escrito el criterio de búsqueda, haremos *clic* en el botón **Encontrar** o bien pulsaremos la tecla Enter.



En primera instancia podemos restringir el ámbito de búsqueda de dos formas. Utilizando el menú desplegable de la parte derecha (donde pone **Español**), podemos elegir el idioma de las páginas a encontrar. En la figura anterior se muestra el resultado tras hacer *clic* sobre dicho texto. Por defecto, se buscan páginas en cualquier idioma. En segundo lugar, utilizando las dos opciones de **Busquedas** podemos restringir la misma a páginas en España o permitir una búsqueda en todo el mundo.

Podemos distinguir tres zonas en los resultados obtenidos con las búsquedas (véase la siguiente Figura). En la primera zona pueden aparecer los sitios patrocinados relacionados con el criterio de búsqueda. En una segunda zona, los resultados de la búsqueda se presentan como una lista con las primeras páginas encontradas. *Altavista* utiliza un sofisticado criterio de ordenación para presentar en primer lugar las páginas más relevantes con respecto al criterio de búsqueda utilizado. Por ejemplo, se presentarán en primer lugar las páginas en las que la palabra o palabras buscadas aparezcan en el título de la página. Cada elemento de la lista contiene en primer lugar un enlace con el título de la página, a continuación una breve descripción de su contenido y la dirección de la misma.



El enlace **Más páginas de …** asociado a cada elemento de la lista, nos permite acceder a diversas páginas relacionadas situadas en el mismo servidor. Si utilizamos el enlace comentado, se listarán otras páginas relacionadas situadas en el mismo servidor. En la parte inferior de la lista aparecen una serie de números de la forma que muestra la siguiente figura. De esta manera, podemos acceder a más resultados obtenidos en la búsqueda: en concreto, se permite acceder a los siguientes grupos de diez páginas encontradas.



En total es posible acceder a las primeras 200 páginas encontradas. Se supone que, dado el criterio de ordenación por relevancia utilizado por *Altavista*, estas primeras 200 páginas son las "más relacionadas" con el tema buscado y normalmente no será necesario acceder a un número mayor. De todas formas, si las páginas encontradas no son satisfactorias, lo mejor es utilizar un criterio de búsqueda más ajustado que permita encontrar un número de páginas más reducido y más acorde con el tema buscado.

Técnicas de Búsqueda en *Altavista*

El número de páginas encontradas utilizando una sola palabra puede ser enorme, sobre todo si realizamos la búsqueda en cualquier idioma y a nivel mundial. Además, muchas de las páginas pueden contener la palabra buscada pero no referida al tema que nos interesa.

- **Idioma.** Una de las formas de restringir el criterio de búsqueda es pedir al buscador que tan sólo encuentre las páginas cuyo contenido se encuentre en un idioma dado. Para elegir el idioma deseado utilizaremos el menú desplegable incorporado en el formulario.

■ **Mayúsculas y minúsculas.** Si escribimos una palabra en el formulario, *Altavista* encontrará tanto las páginas que la contengan escrita con cualquier combinación de mayúsculas y minúsculas. Por ejemplo, si buscamos *asimov*, encontrará las páginas que contengan asimov, Asimov, ASIMOV, e incluso aSimOv.

■ **Acentos y caracteres especiales.** En la palabra o palabras buscadas podemos utilizar acentos u otros caracteres especiales, como ñ, ç, ü, etc. Si al realizar una búsqueda no utilizamos acentos, *Altavista* encontrará tanto las páginas que contengan la palabra acentuada como sin acentuar.
Por ejemplo, si buscamos *informática*, encontrará las páginas que contengan informática e informática. Si escribimos una palabra acentuada, *Altavista* encontrará tan sólo las páginas que la contengan con el acento.

■ **Comodines.** Imaginemos que queremos encontrar las páginas con palabras que empiecen por el prefijo *calaba*, tales como *calabaza* o *calabacín*. En lugar de escribir todas las posibles palabras, en *Altavista* podemos emplear el carácter comodín `*` (asterisco). Por ejemplo, si utilizamos *calaba** para efectuar la búsqueda, encontraremos todas las páginas que hacen referencia a palabras que comiencen por *calaba*.
Sin embargo, debemos andar con cuidado, ya que el buscador dará preferencia mayoritariamente a aquellas páginas en las que se encuentre el prefijo exacto (*calaba* en nuestro ejemplo). Y aunque parezca increíble puede ocurrir que haya bastantes páginas con dicho prefijo (prueba a efectuar esta búsqueda y verás lo que ocurre). En definitiva, el carácter comodín, tal y como su nombre indica, equivale a cualquier sucesión de cero o más caracteres que aparezcan en su misma posición.

■ **Búsquedas con varias palabras.** Si utilizamos varias palabras como criterio de búsqueda y las separamos mediante espacios en blanco, *Altavista* encontrará todas las páginas que las contengan todas a la vez, aunque no estén juntas ni en el mismo orden.

■ **Frases exactas.** Si queremos encontrar una página en la que aparezca una frase exactamente como la escribimos, incluyendo el orden de las palabras, las mayúsculas y los acentos, debemos escribirla entre comillas dobles. Por ejemplo, para encontrar El juego de Ender, escrito exactamente de esa forma, debemos escribir “*El juego de Ender”*.

■ **Términos requeridos.** Supongamos que queremos forzar a que aparezca una determinada palabra en las páginas encontradas. Para ello colocaremos el signo + justo delante de la palabra o palabras requeridas (con espacio en blanco antes del + y sin espacio después).

```
+Brin +Bear +Asimov
```

Tal y como funciona *Altavista*, esto es equivalente a escribir las palabras sin el signo más y separadas por espacio en blanco.

■ **Términos excluidos.** Ahora supongamos que queremos encontrar información

sobre las páginas que contengan la palabra Fundación, pero no la palabra Asimov. Para ello, colocaremos el signo – justo delante de la palabra o palabras excluidas (con espacio en blanco antes del – y sin espacio después).

```
Fundación -Asimov
```



■ **Búsquedas avanzadas.** Si hacemos *clic* en el enlace **Búsqueda avanzada** en *Altavista*, podremos acceder a un formulario de búsqueda con un mayor número de opciones que nos permitirán restringir mejor los resultados obtenidos, tal como ilustra la figura anterior.

El modo de búsqueda avanzada permite establecer criterios de búsqueda más complejos, restringiendo al mismo tiempo las páginas encontradas a un intervalo de tiempo determinado o limitando las búsquedas a un determinado *dominio* de Internet.

En primer lugar, la búsqueda avanzada de *Altavista* pide que construyamos nuestra consulta en la parte superior (con el fondo azul). Tenemos la posibilidad de realizar una búsqueda especificando una serie de palabras que deben encontrarse en las páginas a localizar (**Todas estas palabras**). En este sentido, es similar a efectuar una búsqueda empleando el operador lógico AND: permite localizar páginas que contengan varios términos a la vez. Esto

resulta equivalente a efectuar una *búsqueda con varias palabras*. Por ejemplo, si queremos encontrar páginas que contengan los términos *Vor* y *Bujold*, escribiríamos ambas palabras en este cuadro de texto. En las páginas encontradas los términos no tienen porqué aparecer consecutivos ni en el mismo orden.

También podemos especificar, en el cuadro de abajo, la *secuencia exacta* de palabras que queremos buscar, lo cual equivale a buscar *frases exactas* encerrándolas entre comillas, tal como hemos visto en un apartado anterior. Debajo, tenemos un cuadro titulado **cualquiera de estas palabras**. Aquellas palabras que especifiquemos en este cuadro de texto, pueden encontrarse o no en las páginas a localizar. En este sentido, es similar a emplear el operador lógico OR: permite localizar páginas que contengan al menos una ocurrencia de los términos enlazados. Por ejemplo, si queremos encontrar páginas en las que aparezcan los términos *Barrayar* o *Cetaganda* escribiremos estas palabras en el cuadro de texto indicado.

Finalmente, en el cuadro **y ninguna de estas palabras** podemos especificar aquellos términos que *no* deben aparecer en las páginas buscadas. Esto es similar a especificar palabras con el signo – delante en las búsquedas simples, tal como vimos en el apartado sobre **términos excluidos**, lo cual es similar a emplear el operador lógico NOT: permite localizar páginas que *no* contengan los términos especificados. Por ejemplo, si queremos encontrar páginas relacionadas con *Arthur C. Clarke*, pero en las que no aparezca el término *odisea*, escribiríamos esta última palabra en este cuadro de texto (y el nombre del autor en el cuadro **esta secuencia exacta**).

Además, la búsqueda avanzada de *Altavista* permite restringir la búsqueda de resultados a páginas modificadas en un intervalo de tiempo concreto, que podemos especificar como un determinado lapso transcurrido hasta hoy (**por intervalos de tiempo**) o bien especificando el comienzo y el final de dicho intervalo (**por intervalo de fechas**).

También podemos restringir la búsqueda de maneras más sofisticadas: por el **tipo de archivo**, especificando un formato concreto de entre las posibilidades ofrecidas por la lista desplegable, y por **ubicación**, especificando el dominio concreto de Internet donde debe realizarse la búsqueda (como veremos más adelante, esto también puede hacerse en una búsqueda simple mediante la palabra clave `domain:`) o bien la URL (localización concreta de la página).

■ **El operador NEAR.** El operador `NEAR` permite localizar páginas en las que aparezcan varios términos a la vez y en la que éstos se encuentren próximos entre sí (separados por menos de 10 palabras). De esta forma es probable que la página se refiera a la relación entre ambos términos y no a los mismos por separado.

Por ejemplo, si queremos encontrar las páginas en las que aparezcan próximos el nombre de *Connie Willis* y de su libro *Remake*, usaremos la expresión de búsqueda simple "*Connie Willis" NEAR Remake*.

■ **Búsqueda en campos o atributos.** Normalmente los motores de búsqueda como *Altavista* restringen por defecto las búsquedas mediante palabras clave al texto contenido en las páginas *web*. No obstante, algunos de ellos permiten definir condiciones de búsqueda que permiten encontrar palabras o frases en

determinados campos de la página, tales como el título, la dirección *web* (URL) o los enlaces, entre otros.

En el caso del buscador *Altavista*, podemos realizar este tipo de búsquedas mediante lo que se denominan **funciones especiales de búsqueda**. Se trata de palabras especiales en inglés que pueden añadirse a la condición de búsqueda para restringir la ubicación de las palabras o frases buscadas.

Por ejemplo, la palabra domain: permite restringir las páginas encontradas a un determinado dominio (por ejemplo es, uk, com, edu , …). De esta manera si añadimos al criterio de búsqueda la expresión *domain:edu*, se encontrarán tan sólo páginas en el dominio edu. El uso de esta función especial de búsqueda no sólo está restringido a dominios de primer nivel; prueba, por ejemplo, a efectuar estas dos búsquedas simples en *Altavista*: *Toledo* y *Toledo domain:uji.es*. ¿Qué diferencias observas en los resultados?

3.2.1.2. Google

Si accedemos al buscador *Google*, sin duda el buscador más popular de nuestra época, a través de la URL `http://www.google.es`, podremos ver la pantalla inicial del buscador, que es similar a la captura que muestra la siguiente figura. En la parte superior izquierda de la ventana podemos ver que el buscador contiene diversos paneles que permiten realizar distintos tipos de búsquedas.



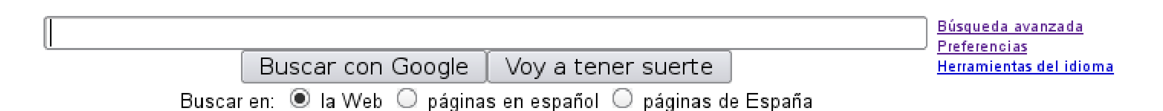
Así, el panel por defecto, **La Web**, permite realizar búsquedas mediante palabras clave utilizando una sintaxis más o menos compleja. El panel **Imágenes** resulta adecuado para realizar búsquedas de ficheros de imagen en la *web*. También podemos efectuar búsquedas (por ejemplo, de calles, de trayectos, etc.) en los mapas de *Google* (*Google Maps*) a través del panel **Maps**.

Si hacemos *clic* en la lista desplegable **Más** podemos acceder a otros paneles interesantes: **Grupos**, que permite efectuar búsquedas en los *grupos de noticias*, **Libros**, para hacer búsquedas dentro del texto de libros o fragmentos de libros digitalizados, y **YouTube**, para acceder y poder hacer búsquedas en uno de los canales de vídeos más populares de Internet.

El buscador *Google* utiliza el cuadro de texto, que podemos ver en la figura que se muestra a continuación, para realizar búsquedas simples. *Google* permite definir, utilizando el enlace **Preferences** (o **Preferencias** en castellano, dependiendo del idioma que se esté utilizando), entre otras cosas, el idioma en el que se van a

visualizar las páginas *web*. Si hemos configurado previamente el navegador para visualizar las páginas en castellano (o catalán, o cualquier otro idioma), se mostrarán directamente en ese idioma si necesidad de acceder a este enlace.

En el recuadro del formulario que muestra la figura escribiremos el criterio de búsqueda, que podrá ser una o varias palabras con algunos símbolos especiales que nos permiten restringir los resultados encontrados. Una vez escrito el criterio de búsqueda, haremos *clic* en el botón **Buscar con Google** o pulsaremos la tecla Enter.



El botón **Voy a tener suerte** accede directamente a la primera página obtenida como resultado de la búsqueda. Podemos restringir el ámbito de búsqueda marcando la opción **páginas en español**, o aún más si seleccionamos la opción **páginas de España** (la primera opción incluiría páginas de, por ejemplo, América del sur que no incluiría la segunda). Por defecto, se buscan páginas en cualquier idioma (búsqueda en la Web).

Tras realizar una búsqueda, como la que ilustra la figura que se muestra a continuación (buscamos *ordenadores portátiles*), se obtiene: una indicación del número de páginas encontradas (182.000 en este caso, tal y como se muestra en dicha figura), la categoría del catálogo más relacionada con la búsqueda, y una relación de las páginas *web* en las que se han encontrado las palabras solicitadas, ordenadas utilizando *PageRank*, un método propio y muy efectivo de *Google*.



El enlace **En Caché**, que igualmente podemos apreciar en la figura anterior, muestra la página *web* tal y como era en el momento en que el buscador la introdujo en su índice. Puede que en la actualidad haya cambiado e incluso que haya desaparecido. El enlace **Páginas similares** muestra distintas páginas que tratan temas parecidos a la encontrada.

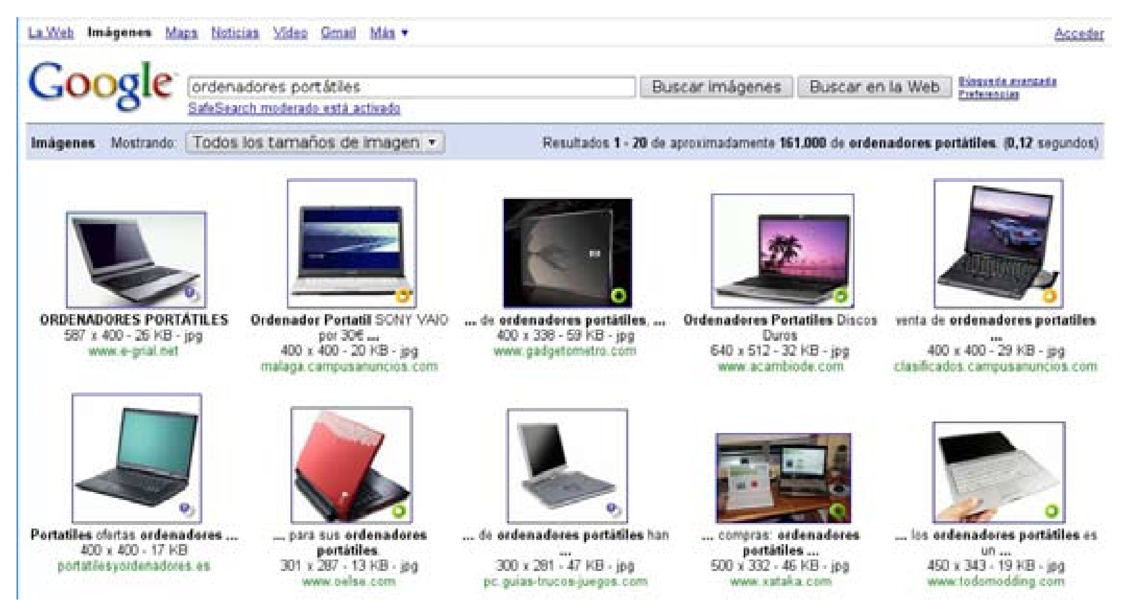
Tal como podemos apreciar en la siguiente figura, en la parte inferior de la ventana que muestra los resultados de la búsqueda aparecen una serie de números

que dan acceso a los siguientes grupos de páginas encontradas.





Si queremos realizar una búsqueda de imágenes en lugar de las búsquedas habituales en la *Web*, tan sólo tenemos que seleccionar el panel **Imágenes** e introducir el criterio de búsqueda en el recuadro del formulario correspondiente. Como resultado de la búsqueda obtendremos un conjunto de imágenes relacionadas con el criterio de búsqueda, junto con un enlace a la página web en donde podemos encontrarlas (véase la siguiente figura).



Tal como podemos apreciar en la figura, en el resultado de la búsqueda también se muestran las características más relevantes de cada una de las imágenes encontradas: sus *dimensiones* (ancho x alto), el *espacio* que ocupa en disco (normalmente en KB) y, si se conoce, el *formato* en el que está almacenada (JPG, PNG,…). Cuando más adelante tratemos de la *búsqueda avanzada de imágenes* comentaremos algunas cosas más al respecto.

Técnicas de Búsqueda en *Google*

El número de páginas encontradas utilizando una sola palabra puede ser enorme, sobre todo si realizamos la búsqueda en cualquier idioma y a nivel mundial. Además, muchas de las páginas pueden contener la palabra buscada pero no referida al tema que nos interesa. Es por ello que conviene estar al tanto de algunas de las técnicas que podemos emplear para restringir mejor nuestras búsquedas en *Google*.

■ **Idioma.** Una de las formas de restringir el criterio de búsqueda es pedir al buscador que tan sólo encuentre las páginas cuyo contenido se encuentre en un idioma dado. Para elegir el idioma deseado utilizaremos el enlace **Preferencias** para elegir el idioma o idiomas, activaremos la opción **Dar preferencia a las páginas escritas en los idiomas siguientes:** y finalmente haremos *clic* en el botón **Guardar preferencias**.

Mayúsculas, minúsculas, acentos y caracteres especiales. Las búsquedas en *Google* no distinguen, *de manera apreciable* entre mayúsculas y minúsculas. Todas las letras, independientemente de como estén escritas, se van a buscar tanto en mayúsculas como en minúsculas. Sin embargo, a la hora de ordenar los resultados se aplicará el *PageRank* de *Google* ya comentado anteriormente. Por ejemplo, buscar: *google*, *GOOGLE* o *GoOgLe* generará básicamente los mismos resultados *relevantes*, aunque puede haber diferencias pequeñas en cuanto al número total de páginas encontradas.

Sin embargo, en *Google* sí que se distinguen acentos, diéresis y demás caracteres "especiales", tales como la letra eñe, a la hora de efectuar búsquedas. No obstante, *Google* considera tanto la versión de la palabra con caracteres especiales como sin ellos a la hora de hacer las búsquedas. Con esto, ocurre algo parecido a lo de usar mayúsculas y minúsculas. Es decir, *Martín* y *Martin* encontrarán un número total de páginas distinto, pero los enlaces de mayor relevancia serán básicamente los mismos. Prueba, por ejemplo, a realizar una búsqueda simple en *Google* con la palabra *desagüe* y otra con la palabra *desague* y observarás las diferencias que ocurren en cuanto al total de páginas encontradas. Sin embargo, hay pocas diferencias entre las páginas localizadas en los primeros lugares.

Se recomienda, no obstante, emplear la forma ortográfica correcta de las palabras con el fin de restringir las búsquedas en la medida de lo posible. Lo normal es que podamos encontrar las páginas buscadas antes si empleamos las formas gramaticalmente correctas. No obstante, si no obtenemos el resultado esperado con las palabras escritas correctamente, podemos probar a escribirlas sin acentos, eñes, etc. para ver si así encontramos los resultados apetecidos.

■ **Comodines.** Tal y como comentan en la ayuda los creadores de *Google*:

"*Para proporcionarle los resultados más exactos, Google no usa "búsquedas parciales" ni realiza búsquedas con "comodines". En otras palabras, Google busca exactamente los términos que ingresa en la casilla de búsqueda. Buscar "sal" o "sal*" no devolverá búsquedas que contengan "salero" o "salamandra". Si tiene dudas, pruebe varias formas distintas: "calle" y "calles", por ejemplo.*"

De lo cual se desprende claramente que *Google* no soporta el uso de comodines.

■ **Búsquedas con varias palabras.** Si utilizamos varias palabras como criterio de búsqueda y las separamos mediante espacios en blanco, *Google* encontrará todas las páginas que las contengan todas a la vez, aunque no estén juntas ni en el mismo orden. De nuevo, citando a los autores:

◆ "*Google sólo le muestra aquellas páginas que incluyen todos los términos de la búsqueda. No es necesario incluir "AND" entre sus términos. Para*

acotar la búsqueda un poco más, agregue más términos."

◆ *"Google ignora las palabras y caracteres comunes, conocidos como términos superfluos. Google automáticamente descarta términos como* `http` *y* `.com`*, así como ciertos dígitos o letras, porque raramente estos términos ayudan a acotar la búsqueda y pueden demorarla en forma significativa."*

◆ *"Use el signo + para incluir términos superfluos en su búsqueda. Asegúrese de incluir un espacio antes del signo +. [También puede incluir incluir el signo + en sus frases de búsqueda.]"*

■ **Frases exactas.** Si queremos encontrar una página en la que aparezca una frase exactamente como la escribimos, incluyendo el orden de las palabras, las mayúsculas y los caracteres "especiales" (tildes, eñes, etc.), debemos escribir todas sus palabras encerradas entre comillas.

■ **Términos excluidos.** Se puede excluir una palabra de la búsqueda colocando un signo menos (**–**) inmediatamente antes del término que se quiere excluir. Es obligatorio dejar un espacio antes del signo menos con respecto a la palabra anterior.

■ **Búsquedas avanzadas.** Si hacemos *clic* en el enlace **Búsqueda avanzada** en la pantalla inicial de *Google*, por ejemplo, accederemos a un formulario de búsqueda con un mayor número de opciones que nos permitirán restringir mejor los resultados obtenidos. Todas las opciones que se pueden utilizar son bastante intuitivas y se muestran en la siguiente figura.



En realidad, muchas de estas opciones son equivalentes a las opciones mostradas por el buscador *Altavista* en su búsqueda avanzada, y que hemos

comentado anteriormente. Al igual que *Altavista*, *Google* pide que construyamos nuestra consulta en la parte superior (con el fondo azul).

Tenemos la posibilidad de realizar una búsqueda especificando una serie de palabras que deben encontrarse en las páginas a localizar (AND) en el cuadro titulado **con todas las palabras** (equivalente al cuadro **Todas estas palabras** de *Altavista*). Al igual que ocurría con *Altavista*, también podemos especificar, en el cuadro de abajo (**con la frase exacta**), la *secuencia exacta* de palabras que queremos buscar, lo cual equivale a buscar *frases exactas* encerrándolas entre comillas.

De la misma manera, podemos buscar **con alguna de las palabras** (OR) y **sin las palabras** (NOT), las cuales son equivalentes a las opciones respectivas de *Altavista* **cualquiera de estas palabras** y **y ninguna de estas palabras**.

Al igual que ocurría con *Altavista* podemos especificar un intervalo de tiempo determinado (pero sólo el lapso transcurrido hasta hoy) y también el formato del archivo buscado entre una serie de opciones predefinidas.

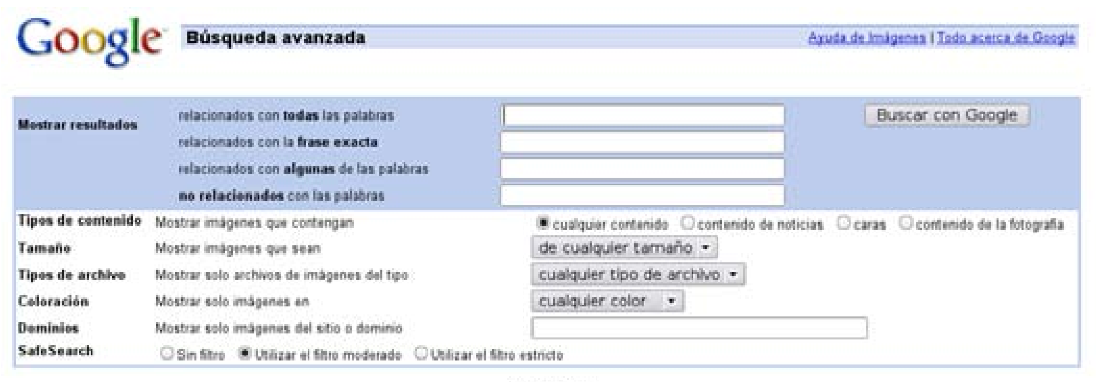
También podemos restringir la búsqueda de maneras más sofisticadas: por **Dominios** (equivalente a la opción **ubicación** de *Altavista*), especificando el dominio concreto de Internet donde debe realizarse la búsqueda (como veremos justo a continuación, esto también puede hacerse en una búsqueda simple mediante la palabra clave `site:`); por el **Idioma** en el que están escritas las páginas y por el país (**Región**) donde se ubica el servidor web en el que se hayan alojadas; o por el lugar de la página (título, contenido, enlaces, …) en el que se encuentran los términos buscados (cuadro **Presencia**).

■ **Restricción de dominio.** Algunas palabras, cuando se añaden con dos puntos, tienen un significado especial para *Google*, al igual que sucedía con *Altavista*. Una de esas palabras es el operador `site:`, equivalente al `domain:` de *Altavista*. Para buscar en un ordenador o dominio específico, se utiliza la sintaxis `site:ejemplodedominio.com` en la casilla de búsqueda simple de *Google*. Por ejemplo, para encontrar información de becas en la UJI, introduciremos:

```
becas site:uji.es
```

en el cuadro de búsquedas simples de *Google*.

■ **Búsquedas avanzadas de imágenes.** También podemos realizar búsquedas avanzadas de imágenes. Para ello debemos estar situados en el *panel de búsqueda de imágenes* de *Google*, y hacer *clic* en el enlace **Búsqueda avanzada**. Una vez realizada esta última acción, accedemos a un formulario de búsqueda similar al que muestra en la siguiente figura, el cual presenta un mayor número de opciones que la búsqueda simple para restringir mejor la búsqueda de imágenes.



El cuadro de fondo azul que se ve en la parte superior permite especificar los términos (palabras) de la búsqueda de manera equivalente a la búsqueda avanzada convencional de *Google* (cuyas opciones ya hemos comentado anteriormente): con todas las palabras, con la frase exacta, con alguna de las palabras, etc. Además, y de manera muy interesante, también permite restringir la búsqueda de manera más sofisticada, mediante la posibilidad de buscar imágenes:

- Por **Tipos de contenido**: si son imágenes de retratos de personas (**caras**), fotografías, etc.

- Por el **Tamaño**: pequeñas, medianas, grandes o extra grande.

- Por el **Tipo de archivo**: formato en el que está almacenado la imagen, como por ejemplo GIF, JPG, PNG, etc.

- Por la **Coloración**: color sólido, escala de grises o blanco y negro.

También, al igual que la búsqueda avanzada convencional, podemos añadir la restricción de **Dominio**, cosa que también puede hacerse con la búsqueda simple de imágenes mediante el empleo del operador `site:`. Prueba, por ejemplo, a realizar una búsqueda simple de imágenes con los términos *Toledo* y *Toledo site:uji.es* y verás la diferencia.

CAPÍTULO 4

4. INTERNET: CORREO ELECTRÓNICO, LISTAS Y FOROS

4.1. INTRODUCCIÓN

El correo electrónico es uno de los servicios más utilizados en *Internet* y permite la comunicación entre personas que tengan acceso al mismo de forma similar a como lo hacemos con el correo tradicional. Para enviar un mensaje, escribimos su contenido e indicamos el destinatario y un sistema de distribución se encarga de depositarlo en un buzón propiedad del destinatario. Debemos comprobar periódicamente nuestro buzón para ver si hemos recibido algún mensaje nuevo.

Los buzones de correo entrante son almacenados en un ordenador central, que denominaremos *Servidor de correo entrante* o *estafeta de correo* y que es capaz de gestionar los mensajes de varios miles de usuarios.

El correo electrónico, al igual que el postal, es privado. Sólo el remitente y el destinatario de un mensaje pueden leerlo. Para garantizar esta privacidad, antes de acceder a nuestro correo debemos identificarnos con nuestro nombre de usuario y una contraseña de acceso (password). Es responsabilidad nuestra recordar siempre la contraseña y, sobre todo, no difundirla.

A la hora de acceder a nuestro correo electrónico, necesitamos un programa cliente que nos permita al menos leer el correo recibido y escribir y enviar nuestros propios mensajes. Existen numerosos programas cliente de correo electrónico y, de hecho, los alumnos de la UJI disponen de varias alternativas a este respecto.

1.- Traer los mensajes recibidos en el ordenador central hasta nuestro ordenador.

- Ventajas:
  - La gestión de los mensajes es muy rápida pues todas las operaciones se realizan en el ordenador local. Sólo se utiliza la red en el momento de transferir los mensajes recibidos desde el ordenador central hasta nuestro ordenador.
  - La cantidad de mensajes que podemos almacenar sólo está limitada por la capacidad de nuestro ordenador que, generalmente, es muy superior al espacio que podemos utilizar en el ordenador central. Ya que el ordenador central tiene que almacenar los mensajes de miles de usuarios, el espacio disponible para cada uno de ellos está limitado.
- Inconvenientes:
  - Si el ordenador local puede ser accedido por otros usuarios, éstos tendrán acceso a nuestros mensajes.
  - Este método es conveniente usarlo cuando dispongamos de un ordenador personal cuyo uso está limitado (por ejemplo, el ordenador de un despacho) o

cuando utilicemos un sistema multiusuario (por ejemplo, UNIX).

- El protocolo que más se utiliza para gestionar el correo de esta forma es el POP3.

2.- Gestionar directamente en el ordenador central los mensajes recibidos.

- Ventajas:
  - Al gestionar los mensajes siempre en el ordenador central, podemos accederlos desde cualquier ordenador.
- Inconvenientes:
  - El espacio disponible para almacenar mensajes está bastante limitado.
  - Todas las operaciones se realizan a través de la red, por lo que la gestión es más lenta.
  - Este método se suele utilizar cuando no se dispone de un ordenador de uso personal (por ejemplo, desde las aulas informáticas).
  - Para gestionar el correo mediante este método se utilizan las pasarelas WebMail.
  - También es posible utilizar el protocolo IMAP4.
  - Se puede utilizar algún cliente de correo (como `mutt` o `pine`) desde una conexión `telnet` o `ssh` al servidor de correo de los alumnos. En este caso, el manejo del correo no se realiza mediante ventanas, menús y botones, sino mediante el uso de combinaciones de teclas.

4.2. EL PUNTO DE ACCESO ÚNICO DE LA UJI (SSO, SINGLE SING-ON)

La UJI ha implantado recientemente el punto de acceso único (http://sso.uji.es). Permite el acceso a cualquier aplicación que requiera la identificación de usuario o el acceso a zonas privadas para consultar el correo, los archivos protegidos, etc. El usuario necesita sus credenciales para identificarse. Estas credenciales son el nombre de usuario y el password o bien el certificado digital.

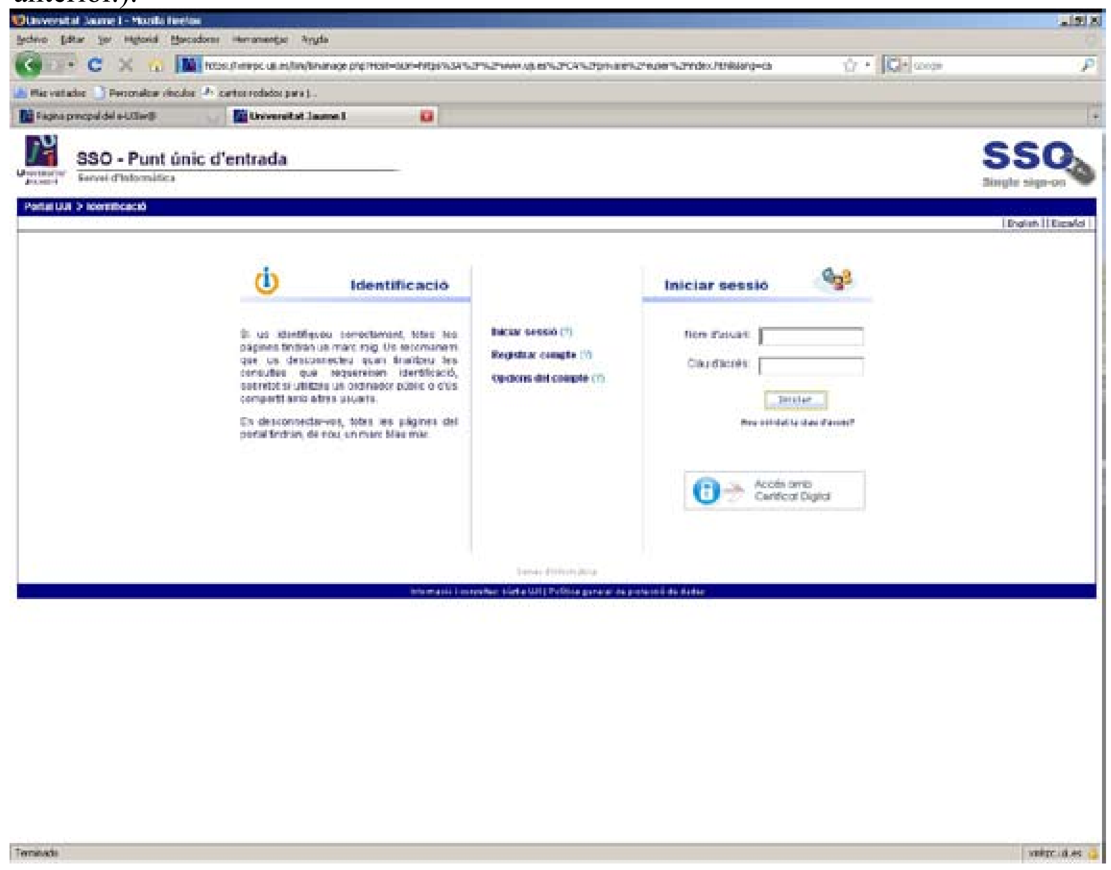
El usuario únicamente tiene que autentificarse una vez y tendrá acceso a cualquier aplicación que necesite las credenciales (aulavirtual, webmail, e-ujier, etc.) . Una vez autentificado se podrán acceder a otras aplicaciones sin necesidad de introducir de nuevo los credenciales. Sólo es necesario introducirlas de nuevo cuando se cierre la sesión activa en alguna de las aplicaciones o cuando durante la sesión se limpien las cookies. Para obtener más información consulta en http://sso.uji.es.

En la siguiente página se muestra una imagen con la pantalla de acceso único de la UJI.

4.3. GESTIÓN DE CORREO CON *WEBMAIL*

Para poder acceder a nuestra cuenta de correo utilizando *WebMail*, debemos arrancar el navegador *Mozilla Firefox* (o cualquier otro) y conectarnos a la página asociada al mismo (`http://webmail.uji.es/`). Si la conexión se realiza correctamente, aparecerá la siguiente ventana, que tras reconocer el navegador la autoridad certificadora de la UJI (hacer clic sobre la flecha azul y los botones de confirmación de los cuadros de diálogo que aparecen), aparecerá una nueva ventana como la que se muestra en la sección anterior.

También se puede acceder desde la página principal de la UJI, (`http://www.uji.es`), accediendo al enlace `WebMail` que aparece en la parte derecha de la página, dentro de la cajita **Eines**, aunque, en este caso, accedemos directamente a la pantalla de identificación de usuario que se muestra en la página anterior.).



4.3.1. Apariencia de WebMail

La primera pantalla que nos aparece al acceder a WebMail, sirve para que nos conectemos a nuestra cuenta de correo en la UJI. Para ello es necesario proporcionar nuestro nombre de usuario y nuestra contraseña tal y como se ha indicado en la introducción. Una vez tecleados estos datos, basta con pulsar el botón Accedir y, si se han introducido correctamente, se accede, en este caso, a la pantalla de gestión de correo.

En la parte superior podemos distinguir distintos botones que nos permitirán

acceder a las distintas funciones del programa:

INBOX: Nos permite acceder a la carpeta de entrada de correo. Tal y como veremos en un apartado posterior, el correo puede clasificarse en carpetas. Esta opción accede a la carpeta en la que se almacena por defecto el correo recibido.

Nuevo: Permite abrir la ventana de redacción de un nuevo correo a enviar.

Carpetas: Permite gestionar las carpetas donde se guarda el correo.

Opciones: Permite personalizar el funcionamiento del programa WebMail.

Buscar: Permite buscar mensajes en base a múltiples criterios.

Ayuda: Permite acceder a la ayuda del programa WebMail.

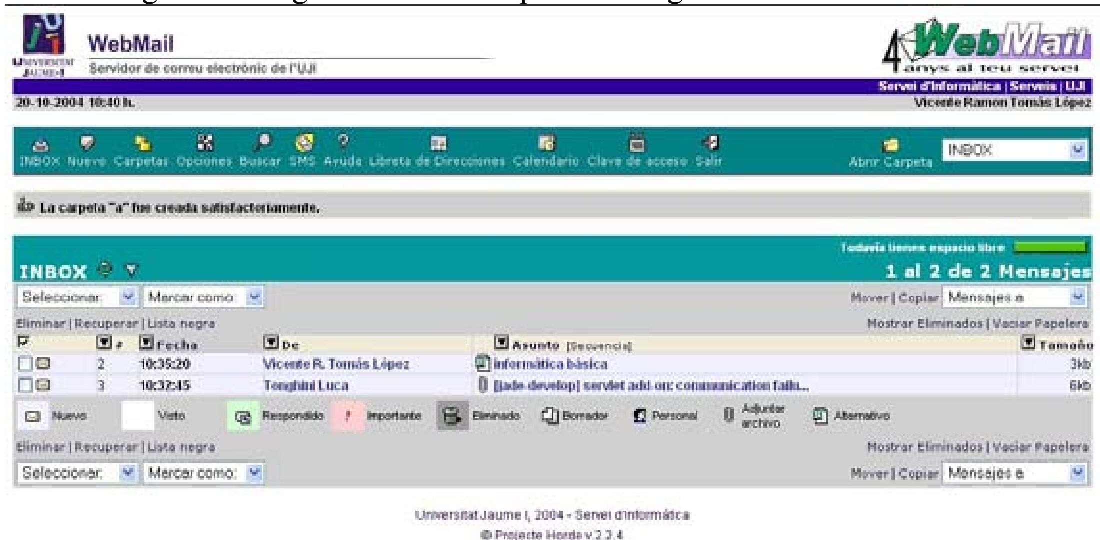
Direcciones: Permite gestionar una agenda de direcciones de correo.

Calendario: Permite gestionar un calendario y agenda.

Salir: Sale del programa WebMail. Por razones de seguridad, es recomendable salir siempre pulsando este botón, en lugar de navegar a la página anterior, cerrar la ventana o salirnos del navegador.

Abrir: Permite acceder a las distintas carpetas de correo.

En la siguiente imagen se observa la pantalla de gestión de correo:



La zona central de la ventana muestra una lista con los mensajes contenidos en la carpeta seleccionada en cada momento. En este caso la carpeta seleccionada es INBOX. En la parte superior derecha aparece una barra la cual nos indica el espacio utilizado, pulsando con el ratón sobre la barra podemos acceder a más información sobre la capacidad de la cuenta.

En la figura anterior podemos ver que la carpeta INBOX contiene dos mensajes, cada uno de los cuales aparece en una línea distinta, y que la información asociada a los mismos se distribuye en una serie de columnas estándar:

La primera columna permite seleccionar una serie de mensajes para borrarlos, moverlos, etc. Además, muestra gráficamente el estado del mensaje según los siguientes iconos:

- : mensaje nuevo, no leído.
- : mensaje respondido.

- : mensaje personal (no dirigido a múltiples destinatarios).
- (Ningún icono): mensaje ya leído.

Los diferentes iconos pueden combinarse entre sí.

- #: Muestra un número que marca el orden en que se recibieron los mensajes.
- **Fecha**: Muestra la fecha y hora en la que se envió el mensaje.
- **De**: Muestra la identidad del remitente.
- **Asunto**: Muestra una breve descripción del contenido del mensaje tal y como ha sido definida por parte del remitente.
- **Tamaño**: Muestra el tamaño del mensaje.

Pulsando sobre la cabecera de estas columnas, ordenaremos los mensajes mostrados en función de la misma. El icono situado a la izquierda de cada cabecera nos indica si la ordenación es ascendente o descendente.

Además, las líneas correspondientes a los diferentes mensajes pueden tener colores distintos en función de ciertas características: **Nuevo**, **Visto**, **Respondido**, **Importante** y **Eliminado**.

- Los pequeños recuadros situados en la parte izquierda de la primera columna nos permiten seleccionar mensajes para aplicarles diversas acciones. Además, podemos seleccionar mensajes y marcarlos con ciertas características (**Visto**, **Respondido**, **Eliminado**, etc.) usando el submenú **Marcar como**:. Una forma más potente de seleccionar grupos de mensajes según diversos criterios es utilizar el submenú **Seleccionar.**

Cada vez que nos conectamos a WebMail se listan los mensajes recibidos hasta ese momento. Si durante la misma sesión queremos comprobar si hemos recibido algún mensaje nuevo, deberemos pinchar sobre el icono **refrescar**, situado al lado del nombre de la carpeta, aunque el programa nos avisa de vez en cuando de la llegada de los mensajes.

Antes de pasar a ver cómo leer y escribir mensajes, veremos cómo salir del gestor de correo. Para ello debemos pulsar el icono **Salir** situado en la parte superior de la página.

4.3.2. Envío de mensajes

Para poder enviar un mensaje comenzaremos por pulsar sobre el botón Nuevo de la parte superior de la página. Como resultado se abrirá una ventana de redacción de mensaje como que se muestra a continuación.



La versión actual de WebMail nos permite utilizar varias identidades con las que enviar correo. Cada identidad se caracteriza fundamentalmente por un nombre de usuario y una dirección de correo. El submenú **Identidad** nos permitirá elegir aquella con la que queramos enviar el mensaje.

Además, será necesario rellenar algunos de los campos de la cabecera del mensaje. En concreto, en el campo **Para** debemos escribir la dirección de correo del destinatario del mensaje y en el campo **Asunto** debemos elegir un texto breve que nos permita definir su contenido.

El campo **Cc** (“Carbon Copy" o "Copia carbón"), se utiliza cuando queremos que otras personas además del destinatario principal reciban también el mensaje. El campo **Bcc** ("Blind Carbon Copy" o "Copia carbón oculta”) funciona de modo similar al anterior, pero en este caso las direcciones utilizadas quedarán ocultas al resto de receptores. Por ejemplo, si en el mensaje de la figura anterior hubiésemos escrito `aznar@la-moncloa.es` en el campo **Bcc**, tanto `bush@guaithouse.gov` como `sadam@irak.gov` sabrían al recibir el mensaje que se ha enviado una copia al otro, pero no sabrían que además hemos enviado una copia a `aznar@la-moncloa.es`.

Si queremos enviar una copia del correo a más de dos personas, basta con escribir sus direcciones separadas por comas en el campo **Para**:, **Cc** o **Bcc**.

La finalidad de los campos **Adjuntar archivos** y **Opciones** se verá en apartados posteriores.

En el recuadro central de la ventana escribiremos el texto del mensaje a enviar y

para hacer efectivo el envío bastará con pulsar el botón **Enviar mensaje**. El botón **Guardar Borrador** coloca una copia del mensaje sin terminar en la carpeta **drafts** (borradores). De este modo, podemos acceder posteriormente al mismo para finalizarlo. Finalmente, el botón **Cancelar mensaje** finaliza el proceso de redacción del mensaje sin enviarlo.

4.3.3. Recepción y contestación de mensajes

Para poder leer alguno de los mensajes recibidos, bastará con pulsar sobre el enlace del asunto o remitente. En ese momento, la lista de mensajes será sustituida por el contenido el mensaje elegido, tal y como se muestra en la siguiente pantalla:



En la parte superior de la página podemos ver la carpeta y la posición del mensaje entre todos los que contiene ("1 de 4" en el ejemplo). Justo debajo aparecen una serie de enlaces que nos permiten realizar diversas operaciones con el mensaje mostrado. Por debajo se muestra ya el contenido del mensaje, dividido en la cabecera y el cuerpo del mensaje. En la parte inferior se repiten los enlaces con las operaciones.

En la cabecera del mensaje se muestran diversos campos con información sobre el mismo, tales como la fecha y hora de envío (**Fecha)**, el remitente (**De**), el destinatario (**Para**), el asunto (**Asunto**) y, en el caso en que contenga algún adjunto, las partes de que consta el mensaje (**Partes**).

El cuerpo del mensaje nos muestra el texto recibido.

Supongamos ahora que queremos responder a un mensaje recibido. Para ello se abre el mensaje tal y como hemos comentado y se elige la opción **Responder** pulsando el enlace correspondiente. El resultado será una nueva ventana de composición de mensaje como la siguiente:

Al elegir la opción **Responder**, el campo **Para** se rellena automáticamente con la dirección de correo del emisor del mensaje al cual contestamos, en este caso *Bill Gates*. También se rellena automáticamente el campo **Asunto** con el texto del asunto del mensaje al que respondemos precedido por el sufijo **Re:** (Respuesta).

Además, al principio del cuerpo del mensaje aparece una línea de la forma

"Mensaje citado por: Bill Gates <bill@microsoft.com>". Esto nos indica que lo que sigue es el contenido del mensaje al que respondemos, que en este caso es el que nos envió *Bill Gates*. A continuación aparece el texto citado de dicho mensaje con sus distintas líneas precedidas por un carácter especial; en este caso el carácter ">". En cualquier punto del mensaje podemos añadir nuestra respuesta, que en el ejemplo anterior se corresponde con las líneas no precedidas por el carácter especial.

De nuevo bastará con pulsar el botón **Enviar Mensaje** para hacer efectivo el envío.

Además de la opción **Responder**, WebMail nos ofrece otras tres acciones que podemos realizar en respuesta a un mensaje recibido:

Responder a todos: es similar a **Responder**, pero en este caso se envía la respuesta a todos los receptores del mensaje original. Recordemos que los mensajes podían ser enviados a más de un destinatario sin más que usar adecuadamente los campos **Para**, **Cc** y **Bcc**.

Reenviar: nos permite enviar un mensaje recibido a una tercera persona, es decir, no respondemos al emisor, sino que lo enviamos a otra dirección que deberemos especificar. Además, nos permite añadir información al mensaje reenviado.

Redirigir: es similar a la opción anterior, pero en este caso no se nos permite modificar el texto del mensaje reenviado, sino que se envía una copia completa del mismo y tan sólo se nos pide la dirección del destinatario.



Una vez leído un mensaje, para volver al listado de los mismos basta con pulsar el enlace Regresar a INBOX. Las dos pequeñas flechas que aparecen a la derecha de este enlace nos permiten desplazarnos al mensaje anterior y siguiente en los contenidos en dicha carpeta.

4.3.4. Borrado de un mensaje

Para borrar un mensaje es necesario hacerlo en dos etapas. En primer lugar colocamos el mensaje en una carpeta especial denominada **Trash** y en segundo lugar vaciamos la papelera.

Podemos colocar los mensajes en la papelera de dos formas. En primer lugar

podemos seleccionar una serie de mensajes a borrar activando el pequeño recuadro que aparece a la izquierda de cada uno en la lista de mensajes. Una vez seleccionados los mensajes, si pulsamos el enlace **Eliminar** de la parte superior de la página, todos ellos se colocarán en **Trash**, se cambiará su color, tachándose la línea correspondiente y se les añadirá un icono especial en la lista indicando su nueva situación.

La otra forma de colocar un mensaje en la papelera consiste en abrirlo tal y como indicamos en el apartado anterior y pulsando el enlace **Eliminar**.

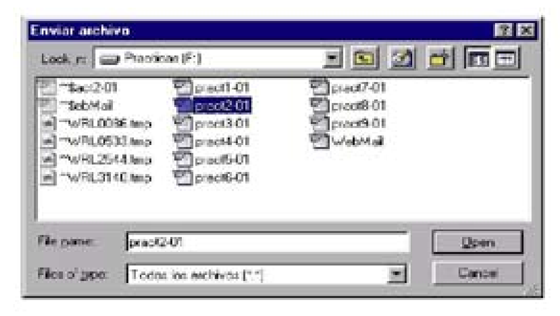
Una vez colocados los mensajes a borrar en la papelera, aún podemos recuperarlos sin más que pulsar el enlace **Recuperar** que aparece sobre la lista. Si queremos borrarlos de modo definitivo, pulsaremos el enlace **Vaciar Papelera**.

4.3.5. Envío de adjuntos

Hasta no hace mucho tiempo, el contenido de los mensajes se limitaba a texto sin ningún tipo de formato, e incluso, podía causar problemas el envío de texto acentuado. En la actualidad es posible enviar mensajes conteniendo texto con formato y en diversos colores, imágenes, páginas web e incluso animaciones y sonidos. Antes de lanzarnos alocadamente a enviar mensajes en tecnicolor y panavisión debemos plantearnos si realmente es necesario. Probablemente la inmensa mayoría de los mensajes que podamos enviar tan sólo deberían contener texto. Además, si nos limitamos a enviar texto, evitaremos los problemas de compatibilidad de formatos entre los distintos programas de gestión de correo, aliviaremos la carga de la red e incluso podemos evitarnos enviar o recibir algún que otro virus.

En todo caso, una de las principales ventajas de los modernos programas de gestión de correo es su capacidad de manejar *Adjuntos*. Un adjunto no es más que un fichero que enviamos junto con el texto del mensaje. Cada mensaje puede contener varios adjuntos y éstos pueden ser de distintos tipos. Así, podemos enviar ficheros generados con un procesador de textos, ficheros gráficos, hojas de cálculo e incluso ficheros con películas o canciones.

Para poder enviar un fichero adjunto tendremos que tenerlo localizado en el disco duro o disquete. En la ventana de composición de mensajes que vimos en el apartado de envío de mensajes, pulsaremos el botón **Navegar** (**Browse** o **examinar** en algunos navegadores), lo que abrirá un cuadro de diálogo como el siguiente y nos permitirá localizar y seleccionar el fichero a adjuntar. Es conveniente seleccionar la opción **Todos los archivos** en el menú **Tipo de archivo** para poder ver en la ventana todo tipo de archivos. En este ejemplo hemos localizado el fichero `prac2-01` en el disco `F:`.



Una vez seleccionado el fichero pulsando el botón **Abrir**, éste aparecerá en el

campo **Adjuntar archivo** tal y como se muestra en la siguiente figura. Para que se adjunte el fichero seleccionado pulsaremos el botón **Adjuntar**.



Por cada uno de los ficheros a adjuntar repetiremos los pasos anteriores, lo que irá incrementando la lista de archivos adjuntos.

Antes o después de adjuntar los ficheros, redactaremos el texto del mensaje, rellenaremos los campos adecuados de la cabecera y lo enviaremos tal y como se vio en un apartado anterior.

Además de enviar adjuntos al enviar un mensaje nuevo, también es posible hacerlo al contestarlo o reenviarlo.

Cuando abrimos un mensaje conteniendo adjuntos, éstos aparecen en la sección **Partes** tal y como puede verse en el siguiente ejemplo:



El correo del ejemplo contiene tres partes. La primera es el propio texto del mensaje, la segunda (2) es un documento Word denominado `pais.doc` y la tercera (3) un documento *pdf* denominado `tema1.pdf`. Si pulsamos el enlace asociado al nombre de un adjunto, el navegador puede comportarse de distinta forma según los programas instalados en el ordenador. En algunos casos visualizará el fichero en una ventana del navegador, en otros abrirá el fichero con otro programa y en otros lo guardará en el disco duro o nos dará la opción de visualizarlo o guardarlo. Por otro lado, si pulsamos el enlace asociado al icono con forma de disquete, se guardará (descargará) el fichero en el disco duro sin visualizarlo.

4.3.6. Gestión de carpetas

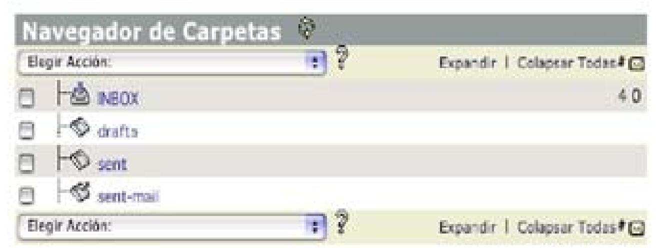
Los mensajes recibidos aparecen por defecto en la carpeta o buzón **INBOX**. Además, si hemos enviado algún mensaje y no hemos modificado las opciones por defecto, tendremos una copia de todos los mensajes enviados en la carpeta S**ent**. Si nos fijamos en alguna de las ventanas de composición de mensaje anteriores, podremos ver que entre las opciones proporcionadas aparece una denominada **Guardar mensaje en la carpeta Sent**. Si dejamos activada esta opción, se guardará una copia de los mensajes enviados en una carpeta con ese nombre. Esto puede ser útil para tener constancia de ciertos mensajes importantes, pero puede llenar nuestro buzón si lo usamos en todos los casos (por defecto, está activa).

Volviendo al tema de las carpetas, podemos visualizar el contenido de cualquier carpeta, seleccionando su nombre en el menú que aparece en la parte superior derecha de la página.



Cuando manejamos unos pocos mensajes podemos tenerlos todos almacenados en la carpeta de entrada **INBOX**. Sin embargo, cuando queremos tener almacenados decenas o centenares de mensajes, nos conviene tenerlos clasificados en distintas carpetas, al igual que lo hacemos con el sistema de ficheros del sistema operativo.

Para gestionar las carpetas con los mensajes, pulsaremos el icono **Carpetas** en la parte superior de la página. La ventana resultante mostrará una lista de nuestras carpetas:



Los recuadros de la izquierda, junto con el submenú **Elegir Acción** nos permiten gestionar una estructura de carpetas para clasificar nuestros mensajes. Antes de realizar cualquier acción con una carpeta hay que seleccionarla haciendo un clic en la casilla de su izquierda. Las acciones básicas que podemos aplicar son las siguientes:

- **Crear carpeta:** crea una carpeta nueva donde podemos almacenar nuestros correos.
- **Renombrar carpeta:** cambia el nombre de una carpeta seleccionada.
- **Eliminar carpeta:** borra la carpeta seleccionada. Antes de borrarla, si contiene algún mensaje el programa nos pide que confirmemos el borrado, puesto que se eliminarán tanto la carpeta como todos los mensajes que contenga.
- **Vaciar carpeta:** borra todos los mensajes que contienen las carpetas seleccionadas.
- Para poder mover o copiar mensajes entre carpetas, los seleccionamos activando el pequeño recuadro situado a su izquierda, elegimos la carpeta destino mediante el submenú de la siguiente figura y elegimos la acción a realizar: **Mover** o **Copiar**.

4.3.7. Uso de la libreta de direcciones

Ya sabemos que para enviar un mensaje a alguien, basta con conocer su dirección de correo electrónico. Sin embargo, cuanto más utilizamos el correo electrónico, más nos damos cuenta de que resulta muy difícil recordar muchas de las direcciones y, en el mejor de los casos, es siempre tedioso tener que escribir largas direcciones

una y otra vez.

Para aliviar algo este problema se puede utilizar la libreta de direcciones. Esta libreta funciona como una agenda telefónica, pero almacena la información y direcciones de correo electrónico de las personas o instituciones con las que mantenemos contacto. Además, la misma herramienta nos permite acceder a un servicio de directorio corporativo (en el ámbito de la UJI) llamado **LDAP**, lo que nos permite encontrar fácilmente las direcciones de correo del personal y de los estudiantes de esta universidad.

Podemos gestionar nuestra libreta de direcciones mediante el botón **Libreta de Direcciones** que aparece en la parte superior de la ventana principal.



En la parte superior aparecen todas las opciones disponibles para la gestión de la libreta de direcciones. Vamos a revisar las fundamentales:

- **Examinar:** Permite listar el contenido de nuestra libreta de direcciones.
- **Agregar:** Permite añadir nuevas direcciones a la libreta.
- **Buscar:** Permite buscar direcciones de correo utilizando un formulario simple.
- **Búsqueda avanzada:** Permite buscar direcciones de correo utilizando un formulario un poco más completo.
- **Importar/Exportar:** Permite importar o exportar listas de direcciones que hayamos creado con otros programas cliente de correo, p.ej. Netscape Messenger.
- **Opciones:** Permite personalizar ciertas opciones de la lista de direcciones.
- **Ayuda:** Debería proporcionar una pequeña ayuda sobre el funcionamiento de las distintas opciones.
- **Correo:** Permite volver a la ventana principal de WebMail.
- **Calendario, Memorandums y Tareas:** Estas opciones permiten gestionar una agenda personal, fijar tareas para determinados días, recibir avisos de modo automático, etc.
- **Salir:** Finaliza la sesión actual de WebMail

El submenú **Buscar** nos permite realizar la búsqueda a partir del nombre, la dirección de correo o la extensión telefónica (p.ej. si se trata de algún profesor). La búsqueda puede realizarse en la libreta de direcciones del usuario, o en el directorio (**LDAP**) de la universidad. Por ejemplo, si buscamos por nombre, y escribimos `badia` en el recuadro, el resultado será una lista de alumnos y profesores cuyo nombre o apellidos contengan esa palabra.

Si realizamos varias búsquedas consecutivas, los resultados obtenidos se irán acumulando en la misma lista. Si queremos borrar la lista, debemos pulsar el enlace **Borrar Búsqueda** de la parte superior derecha. Una vez encontrada la lista de resultados, si pulsamos un nombre, podremos ver la información almacenada en la lista de direcciones. Pulsando una dirección de correo podremos enviar un mensaje a

la misma. Para ello se abrirá automáticamente la ventana de composición de mensaje y en el campo **Para** aparecerá la dirección correspondiente.

Además de usar el directorio de la universidad podemos mantener nuestra propia libreta de direcciones, pudiendo incluir también las de personas ajenas a la Universidad. Para añadir una dirección a nuestra libreta, pulsaremos el botón **Agregar**.

Si pulsamos el botón **Examinar** en la ventana de gestión de direcciones, podremos ver el contenido de nuestra libreta de direcciones.

Para poder modificar o borrar alguna de las direcciones, podemos pulsar el nombre asociado y elegir las opciones **Editar** o **Eliminar** respectivamente.

Supongamos que queremos enviar un correo a alguien incluido en nuestra libreta o en el directorio de la UJI. Para ello, desde la ventana principal de WebMail, pulsamos el botón **Nuevo** y abrimos la ventana de composición de mensajes. En cualquiera de los campos **Para**, **Cc** o **Bcc** podemos escribir parte del nombre o dirección de correo del destinatario. Si a continuación pulsamos la opción **Expandir Nombres**, automáticamente se buscará en el directorio **LDAP** de la UJI y en nuestra libreta alguna persona cuyo nombre o dirección contenga el texto escrito. Si sólo hay una, se añadirán automáticamente sus datos en el campo correspondiente. Si hay más de una que cumpla los requisitos, podremos elegir la que queramos en el menú desplegable que aparecerá en el campo correspondiente. Por ejemplo, si escribimos el texto `merche` en el campo **Para**, y pulsamos la opción **Expandir Nombres**, el resultado será el siguiente:



El proceso anterior facilita enormemente identificar a los destinatarios de nuestros correos: No tenemos por qué acordarnos de su nombre completo o de su dirección de correo, basta con que recordemos parte de su nombre o dirección, o bien hayamos usado un nombre corto y fácil de recordar al añadirlo en nuestra libreta de direcciones.

Otra de las opciones asociadas a la libreta de direcciones que podemos usar desde la ventana de composición de mensaje es la que nos permite acceder a la libreta para rellenar los distintos campos de destinatario a partir de ella. Para ello pulsaremos la opción **Direcciones**.

Si escribimos un nombre o dirección en el campo **Encontrar**, podemos buscar a cualquiera que contenga el texto escrito pulsando el botón **Buscar**. Después, podemos elegir a los destinatarios deseados en la lista de la izquierda y añadirlos al

campo deseado (**Para**, **Cc** o **Bcc**) pulsando el botón correspondiente. El botón **Eliminar** permite borrar los destinatarios seleccionados de la parte derecha, mientras el botón **Limpiar** los borra todos.

La pasarela WebMail ofrece otras posibilidades, tales como el establecimiento de filtros para gestionar o borrar de modo automático los correos recibidos. También es posible gestionar una agenda (calendario), tal y como hemos comentado anteriormente. Además, podemos personalizar múltiples características del entorno utilizando el botón Opciones. Estas opciones incluyen aspectos tales como el idioma, la realización de acciones automáticas con el correo, la visualización de los mensajes, etc. Sería necesaria una práctica mucho más amplia y prolongada para introducir todas estas opciones. No obstante, se anima al alumno a practicar con estas posibilidades del programa y a aprender a utilizar aquellas que le parezcan más útiles.

4.3.8. Configuración del cliente de correo

Además de las pasarelas web para leer el correo, también podemos utilizar aplicaciones específicas de correo electrónico, tales como Mozilla ThunderBird, Kmail y Evolution.

Los pasos a realizar cuando entramos por primera vez en una de estas aplicaciones, consisten en configurar tu cuenta de correo, dando la siguiente información:

- Nombre. Éste te identifica en los correos que envíes.
- Dirección de correo electrónico.
- Nombre del servidor de correo entrante (POP o IMAP).
- Nombre del servidor de correo saliente (SMTP).
- Cuenta de usuario y contraseña.

Si quieres configurar una de estas aplicaciones con tu cuenta de correo de la UJI, puedes consultar la página web del Servei d'Informàtica, http://www.si.uji.es, dentro de “Manuals i documentacions” en la sección "Ajudes i manuals en línia".

4.4. LISTAS DE CORREO

Una de las principales aplicaciones de Internet es el libre intercambio de ideas y opiniones. Existen diversos mecanismos para lograr este objetivo en la red. Las listas de correo son una de las formas que permiten este intercambio de información. El objetivo es que diversas personas interesadas en un mismo tema, puedan conversar, intercambiar opiniones o plantear preguntas. Son ejemplos de temas de debate "Cirugía general y del aparato digestivo", "Teoría y práctica sobre la hacienda pública", "Innovación didáctica del derecho" o "Bonsais".

Las listas de correo, también denominadas listas de distribución o listas de discusión, funcionan a través del correo electrónico. En el caso de las listas, es necesario, habitualmente, un proceso de abono previo conocido como suscripción a la lista.

Cada vez que alguien quiere manifestar una opinión, hacer una pregunta, dar una

respuesta, etc. envía un mensaje de correo electrónico a la lista, y éste llega automáticamente a cada uno de los usuarios que se han abonado previamente. Por lo tanto, si un usuario se abona a una lista, recibirá en su buzón de correo todos los mensajes que los abonados envíen a la misma a partir de ese momento. Últimamente las listas de correo se están integrando en la web, con lo que pueden utilizarse mediante un navegador web, visitando la página asociada.

Las listas de correo están normalmente relacionadas con un tema concreto: un problema de empresa, una tecnología, un programa de ordenador, etc. Todo ámbito de reflexión es susceptible de ser objeto de una lista de correo. Todo usuario puede crear una de estas listas, no existe ninguna restricción.

4.4.1. Suscribirse a una lista de correo

Existen unos programas, denominados *gestores de listas*, que se encargan de gestionar conjuntos de listas de correo. Por ejemplo, el servidor de listas de la UJI, *Mailman* gestiona las listas de correo de nuestra universidad. Enviando mensajes de correo electrónico al servidor, se realizan las suscripciones y las bajas en las listas, se obtiene información sobre las listas existentes, etc. También pueden utilizarse parte de sus opciones accediendo a la página web asociada (por ejemplo, `http://nuvol.uji.es/mailman/listinfo/eines`). A continuación, se describen algunas de las órdenes que se pueden utilizar con este servidor en concreto para realizar las distintas acciones. Estas órdenes se indican en el **asunto** o en el **cuerpo del mensaje** de correo electrónico que el usuario debe enviar al gestor. En un mismo mensaje se pueden mandar varias órdenes en líneas diferentes. Las palabras en negrita son obligatorias, mientras las palabras en cursiva son optativas y deben ser sustituidas por un valor concreto. Por ejemplo, en las órdenes que incluyen la palabra contraseña, está debe sustituirse por la contraseña de cada usuario en particular: la asociada a la lista, no a su cuenta de correo.

- **subscribe** *contraseña dirección.* Suscribe a la lista a la que se envía la dirección de usuario especificada utilizando la contraseña dada. Es mejor no utilizar la misma contraseña que en tus cuentas de correo.
- **unsubscribe** *contraseña dirección.* Pide que se dé de baja en la lista la dirección de usuario especificada.
- **who.** Si la lista lo permite, muestra la dirección de todos los usuarios suscritos.
- **info.** Muestra información general sobre la lista.
- **lists.** Muestra las listas disponibles en el gestor.
- **help.** Muestra las órdenes que acepta el servidor.
- **end.** Detiene el procesamiento de órdenes (útil si se mandan los mensajes con firma).

En el gestor Mailman, cada lista tiene dos direcciones de correo asociadas: la utilizada para su gestión (suscribirse, darse de baja, cambiar la contraseña, etc.), y la utilizada para enviar correos a los usuarios suscritos a la misma. En el caso de este gestor en la UJI, la dirección de gestión es `nombre-`

request@llistes.uji.es (p.ej. eines-request@llistes.uji.es), mientras la dirección de envío es nombre@llistes.uji.es (p.ej. eines@llistes.uji.es).

Hay que tener en cuenta que hay varios tipos de servidores de listas (majordomo, listproc, sympa, listserv, maiser) y, por lo tanto, las órdenes pueden ser diferentes en cada uno de ellos.

Una vez hemos recibido los nombres y direcciones de gestión de las distintas listas de correo de un servidor de correo, podemos ya suscribirnos a alguna de ellas. Si nos queremos suscribir a la lista eines de la UJI, debemos mandarle un mensaje en el que escribamos en el cuerpo del mensaje subscribe eines. Esta petición hará que recibamos un mensaje pidiéndonos que confirmemos la suscripción a la lista. Tras confirmar esta suscripción, y si la lista es abierta, seremos dados de alta en la misma y podremos enviar y recibir mensajes al resto de sus integrantes.

4.4.2. Enviar mensajes a una lista de correo

Siguiendo con el ejemplo anterior, una vez suscritos a la lista eines, los mensajes con las intervenciones del usuario se deberán enviar a eines@llistes.uji.es.

En cuanto a la distribución de los mensajes, hay listas *moderadas* en las que el propietario autoriza o deniega el envío de un mensaje a los suscriptores. De este modo se puede evitar, por ejemplo, el reparto indiscriminado de publicidad, o la llegada de mensajes equivocados. Otras listas son *no moderadas*, de modo que todos los mensajes se envían automáticamente a los suscriptores sin que sean revisados.

Tened en cuenta que las listas de correo no son grupos de charla (chats) y no sirven para intercambiar multitud de preguntas y respuestas rápidamente, sino para realizar discusiones de mayor longitud y profundidad y en las que el intervalo entre las consultas y las respuestas suele ser mucho mayor.

Por otro lado, también existen buscadores especializados en listas de correo que permiten acceder a las mismas a través de la web. (por ejemplo, desde la página web http://www.elistas.net)

Las listas se pueden configurar a través de una página web y no enviando mensajes a un servidor de listas. Esto es bastante común en las listas integradas en web. Por ejemplo, todas las listas incluidas en elistas.net pueden manejarse por la web o mediante un cliente de correo sin necesidad de acceder a la web. Así, para darse de alta en la lista de correo de la asignatura (eines), basta con enviar un correo en blanco y sin asunto a la dirección eines-altas@eListas.net y para darse de baja, es necesario enviarlo a eines-baja@eListas.net. Asimismo, para enviar mensajes a la lista sin usar la web, basta con enviarlos a la dirección de la misma: eines@eListas.net.

4.5. FOROS DE DISCUSIÓN - WEBFÓRUM

Últimamente existe una clara tendencia a que los distintos servicios de Internet se integren en la web. Así, en la primera parte de la práctica pudimos gestionar el correo electrónico mediante una pasarela web (WebMail). También las listas de correo y los grupos de noticias están siendo “absorbidas” por la web. De este modo, se integran en nuevos servicios ofrecidos por portales web o proveedores, a menudo

denominados *foros web*.

La UJI acaba de incorporar un servicio de este tipo al que se puede acceder a través de la dirección `webforum.uji.es`. En esta dirección se incluye, por un lado, los grupos de noticias gestionados por el servidor de grupos de la UJI (`news.uji.es`), y por otro lado, una serie de foros web tanto abiertos como cerrados.

Podemos encontrar en Internet buscadores de foros web, aunque estos normalmente están restringidos a encontrar aquellos foros asociados a algún servidor o portal. Ejemplos de estos buscadores son los que puedes encontrar en `foros.ya.com` o `www.forosdelweb.com`.

4.5.1. Acceder a webfórum

Para acceder a la página de foros de la UJI debes conectarte a las siguientes direcciones: `http://webforum.uji.es` o `https://webforum.uji.es`. Es recomendable la utilización de la segunda dirección cuando se trate de foros cerrados, con el objeto de garantizar al máximo la confidencialidad de las comunicaciones y sobre todo de la clave de acceso.

4.5.2. Tipo de foros

Actualmente los foros que proporciona la pasarela webforum se estructuran de la siguiente manera:

- *Abiertos*: El acceso a los mismos es como su nombre indica libre, no hay que autentificarse para acceder a ellos.

- *Restringidos*: En este tipo de foros hay que autentificarse para poder acceder a ellos. (tanto para leer los mensajes como para enviar) La identificación se realiza mediante tu login y tu clave de correo de la UJI.

- Dentro de los grupos *restringidos* existe otra subclasificación:

- Grupos C*omunitarios*: Pueden acceder únicamente los miembros de la comunidad universitaria

- Grupos *Privados*: Son grupos gestionados por un responsable, que es el encargado de la administración del foro. Para poder acceder al foro es necesario pedir autorización al administrador.

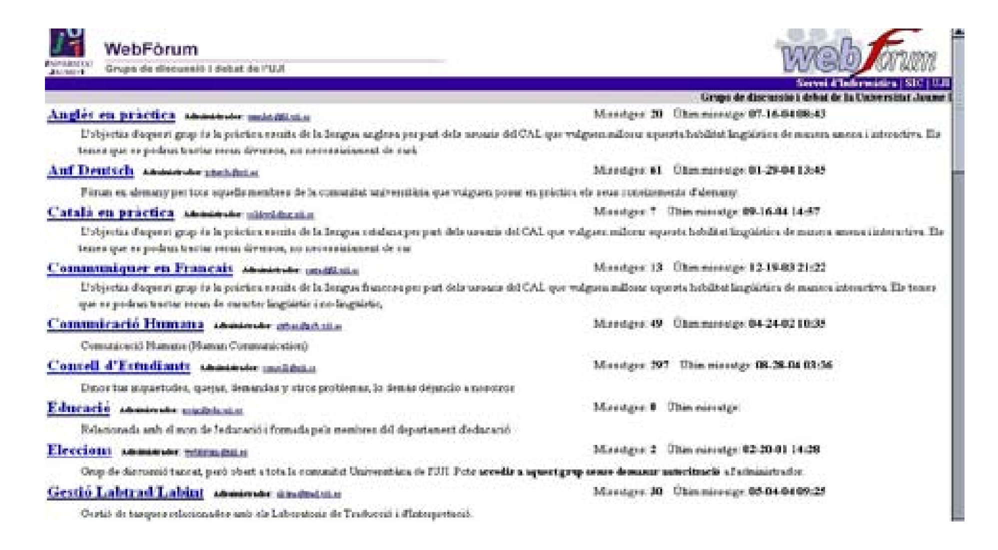
Es importante destacar que en los foros abiertos se puede tener la identidad que se quiera, mientras que en los foros restringidos siempre se esta identificado por el nombre y la dirección de correo real con la que os identificáis al entrar.

4.5.3. Utilización del webfórum

En la siguiente imagen aparece la pantalla de inicio de *webforum.uji.es:*



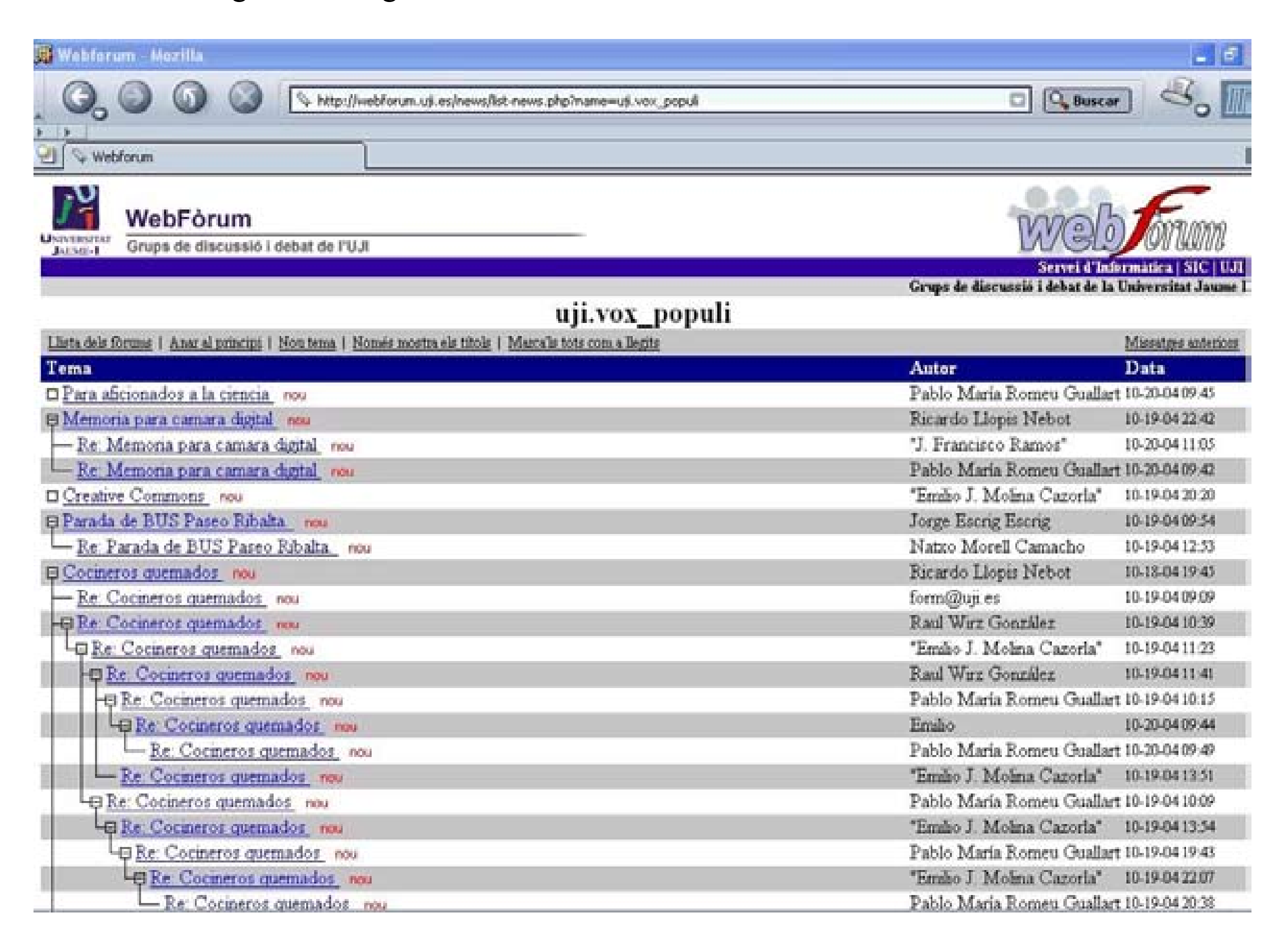
Para ver el listado de los grupos debes pulsar uno de los dos iconos de acceso. Según sea la elección te aparecerá un listado u otro. En el listado, además aparece el nombre del administrador, el número de mensajes que contiene, cuando se ha escrito el último mensaje, y una definición del objetivo del foro. A continuación se muestra el listado de foros restringidos:



Para acceder a un foro en concreto, debes seleccionarlo pulsando con el ratón sobre el. Dependiendo del tipo de foro que se elija, se deberá realizar la autentificación en este momento.



En la siguiente imagen se observa el contenido del foro una vez dentro de él:



En la parte superior de la misma, existe un conjunto de acciones que se pueden realizar dentro del foro. Las acciones son:

- Listado de foros existentes (Llista de fòrums)
- Ir al inicio(Anar al principi)
- Nuevo mensaje (Nou tema)
- Sólo títulos (Només mostra els títols)
- Mostrar todos como leídos (Marca'ls tots com a llegits)

4.6. GRUPOS DE NOTICIAS

Tanto en las noticias, como en las listas de correo, el contenido de la información en tránsito de un emisor hacia varios destinatarios es similar. Sólo cambia el modo de difusión. En las listas de correo el mensaje se envía a un conjunto de buzones personales, mientras que en las noticias el mensaje se almacena en un “gran libro” que puede ser consultado por todos los usuarios de Internet.

Las noticias son gestionadas por servidores que se intercambian varias veces al día las contribuciones aportadas por los usuarios, a fin de que todos los servidores estén casi al día al mismo tiempo. Estas contribuciones se almacenan durante un tiempo predefinido (entre una semana y quince días).

Algunos grupos de noticias son moderados, de modo que los mensajes enviados son leídos previamente por el moderador del grupo, quien decide si esa contribución es compatible o no con la ética del grupo antes de validarla y difundirla.

El servicio de noticias de Internet ofrece la posibilidad de participar en una serie de foros de debate o discusión (denominados grupos de noticias) acerca de diversos temas. Los grupos de noticias se agrupan en jerarquías, cada una de las cuales se dedica a un área de interés particular; por ejemplo, la jerarquía **comp** trata sobre temas relacionados con la informática, mientras que la jerarquía **rec** está dedicada a temas de recreo. En la actualidad, el servidor de noticias de la Universidad Jaume I contiene más de 13000 grupos de noticias.

Por ejemplo, entre los grandes temas en los que se incluyen los distintos grupos de noticias podemos citar los siguientes:

- **comp** : incluye grupos relacionados con la informática.
- **soc** : incluye grupos sobre temas sociales.
- **sci** : incluye grupos sobre ciencia.
- **es** : incluye grupos en castellano.
- **alt** : incluye grupos sobre múltiples temas alternativos.

Los nombres de los temas citados actúan como prefijo de los nombres de los grupos que incluyen. A su vez dentro del tema es, podemos distinguir, los siguientes subtemas: *es.humanidades, es.ciencia, es.comp*, ... Finalmente, dentro de éstos existen otros subgrupos, por ejemplo:

Las contribuciones individuales a un grupo de noticias se denominan artículos. Los diversos temas de debate en que se organizan los artículos dentro de un grupo de noticias se denominan hilos de discusión o cadenas.

4.7. REALIZACIÓN DE ESTA PRÁCTICA

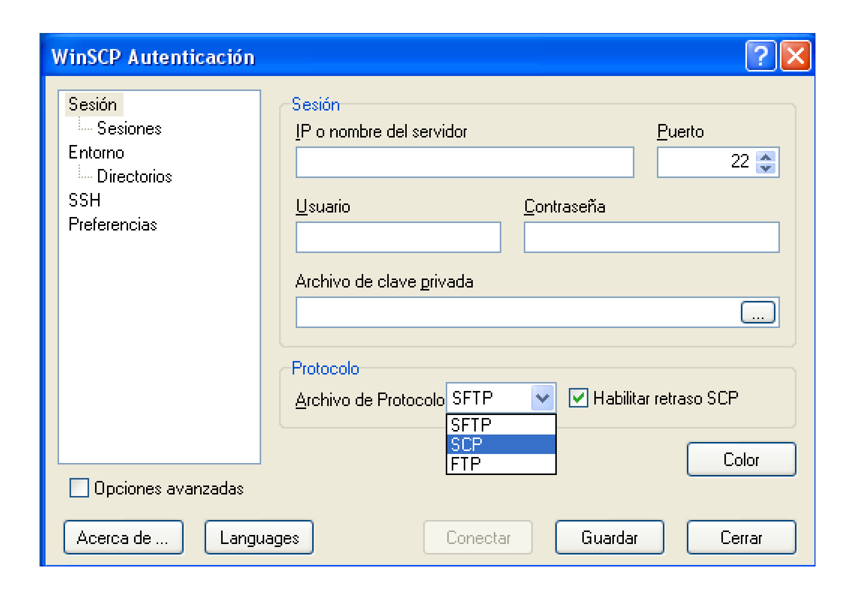
Tanto el boletín de esta práctica como el material de apoyo ha sido diseñado a partir de las prácticas de la asignatura del curso 2001-2002 (`http://terra.act.uji.es/ib/p4`), el "Tutorial básico para la utilización de recursos de red" (`http://marmota.act.uji.es/proyecto-use`) y las prácticas de la asignatura "Herramientas informáticas para la documentación (J42)". (`http://icc2.act.uji.es/j42`).

CAPÍTULO 5

5. INTERNET: SSH, SCP Y FTP

5.1. WINSCP

WinSCP es, básicamente, un cliente de código abierto de SFTP (Secure File Transfer Protocol) y SCP (Secure CoPy) para Windows usando SSH (Secure SHell). Su principal función es copiar archivos entre un ordenador local y un ordenador remoto de forma segura. Además de esta función básica, WinSCP también puede realizar otras operaciones básicas con archivos: renombrar archivos y carpetas, crear nuevas carpetas, cambiar permisos, crear enlaces simbólicos o accesos directos.

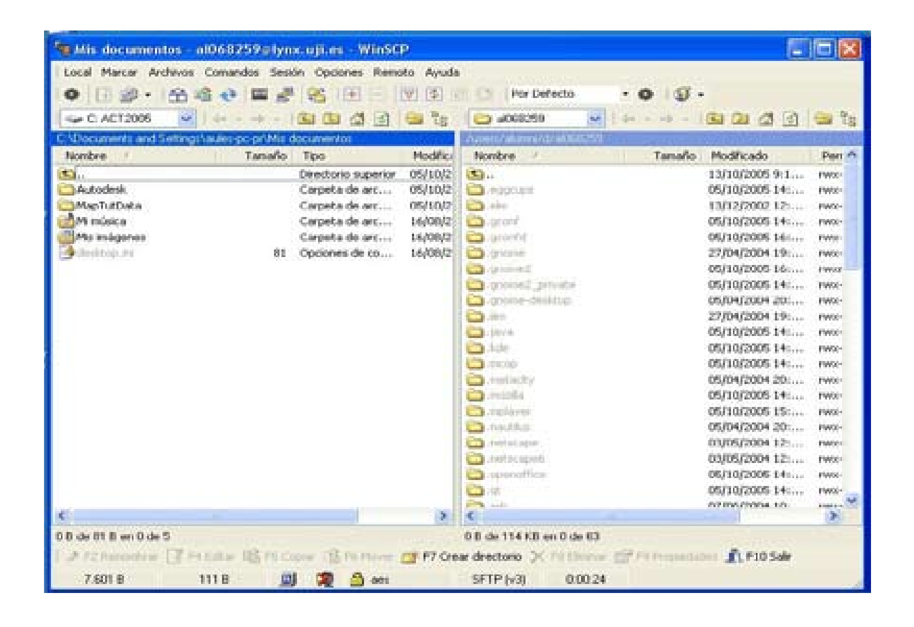


Una vez instalado el programa, al iniciarlo nos aparece una ventana para configurar la conexión con el ordenador remoto, donde tenemos que proporcionar, primordialmente, la IP o nombre del servidor, el nombre de usuario y su contraseña (mejor no teclearla en esta ventana y esperar a que nos la pida el programa en el momento de establecer la conexión), y especificar el protocolo que deseamos utilizar (SFTP, SCP o FTP, con preferencia por las dos primeras opciones).

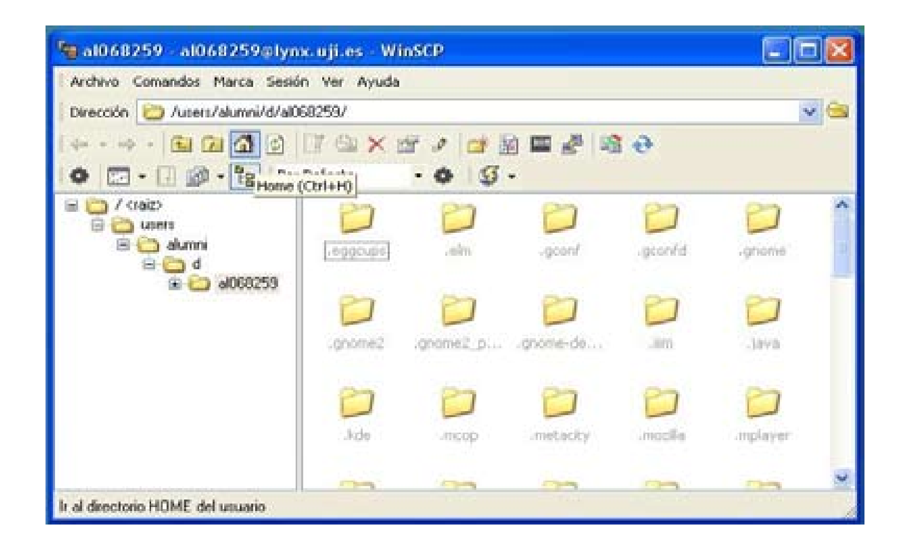
Si queremos guardar esta configuración para utilizarla en posteriores conexiones simplemente tenemos que pulsar en el botón 'Guardar' y darle un nombre que la identifique. Haciendo clic en el botón de 'Conectar' iniciaremos el proceso de conexión.

Al instalar WinSCP se puede elegir el aspecto de la interfaz de usuario que queremos utilizar:

- *Modo Paneles duales o NC (Norton Commander)*: En este modo, disponemos de dos paneles , en el panel de la izquierda se muestran las carpetas y archivos locales, y en el de la derecha los remotos.

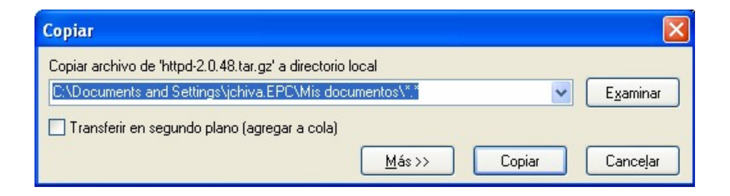


■ *Modo 'Explorador de Windows'*: En este modo solo hay un panel en el que se muestran los archivos del ordenador remoto.



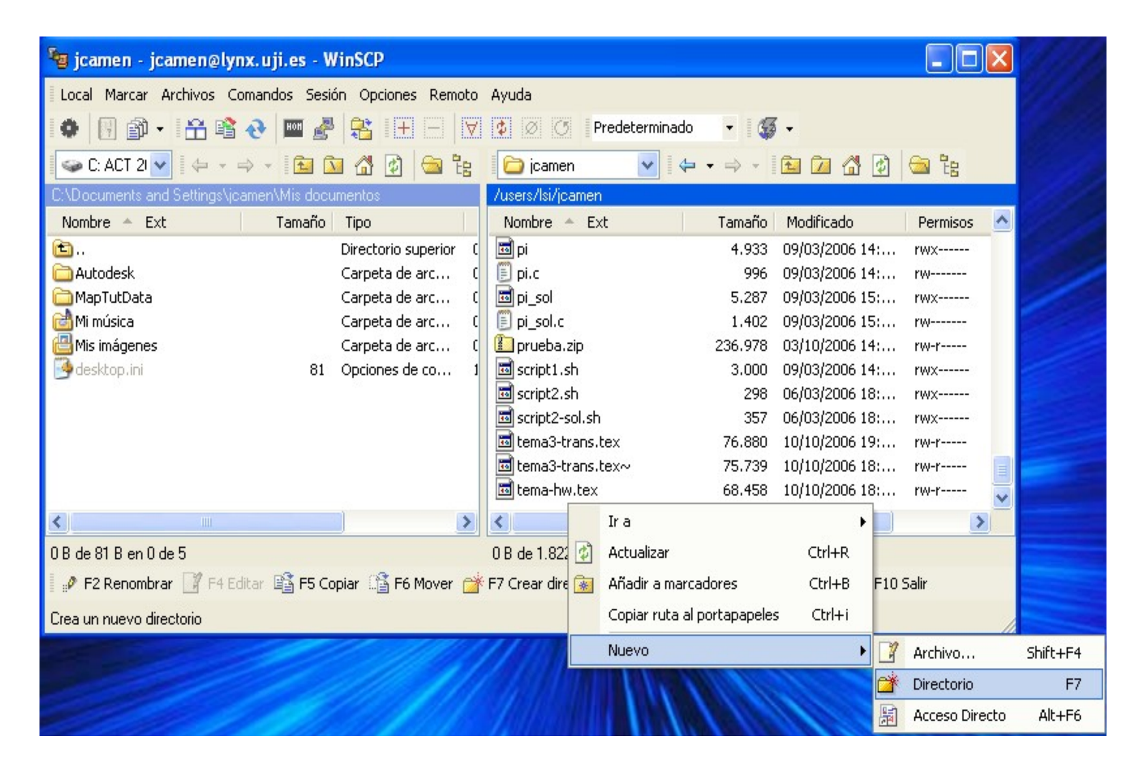
Si queremos copiar un archivo del ordenador remoto al local, basta con seleccionarlo en el panel derecho y arrastrarlo a la carpeta adecuada en el panel izquierdo si estamos en el *modo NC* o, si estamos en modo *Explorador de Windows*, seleccionarlo en el único panel que tenemos y arrastrarlo sobre una carpeta en nuestro ordenador local.

Si queremos copiar un archivo del ordenador remoto al local, basta con seleccionarlo en el panel derecho y arrastrarlo a la carpeta adecuada en el panel izquierdo si estamos en el *modo NC* o, si estamos en modo *Explorador de Windows*, seleccionarlo en el único panel que tenemos y arrastrarlo sobre una carpeta en nuestro ordenador local.

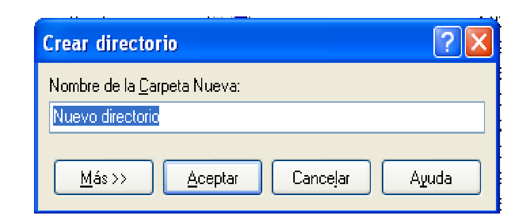


Para finalizar la conexión, pulsaremos sobre el botón 'Salir' de la barra inferior (o pulsando F10) si estamos en *modo NC* o accediendo a la opción 'Salir' del menu 'Archivo' si el modo es el *Explorador de Windows*.

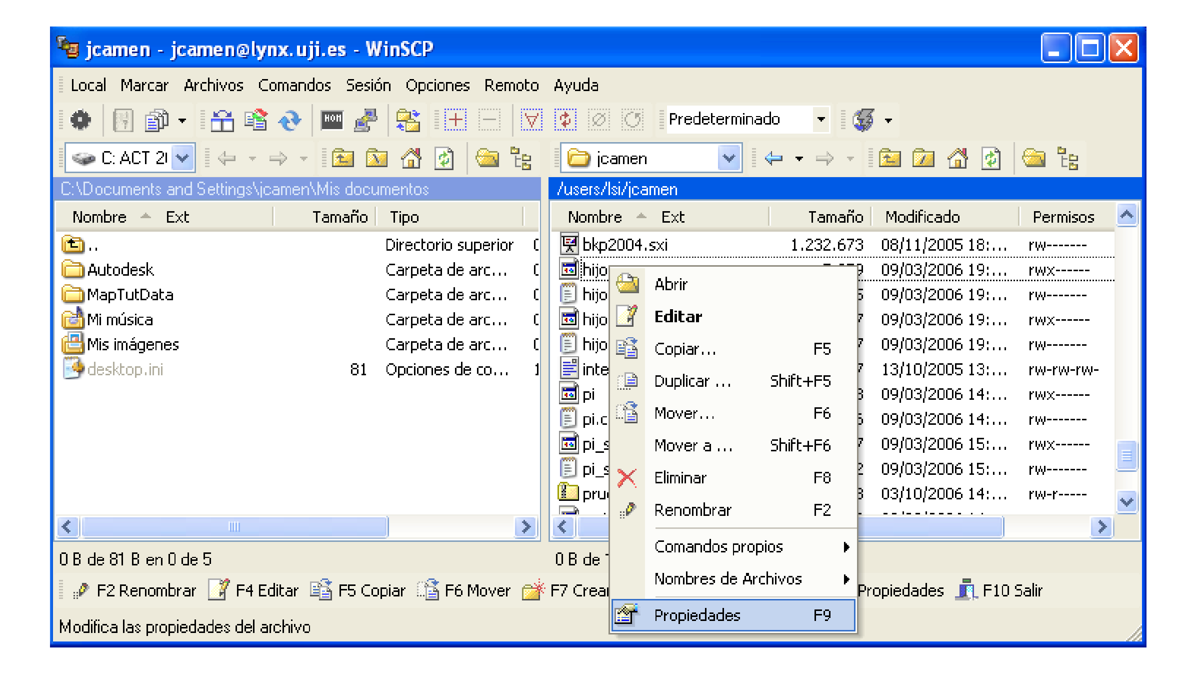
Podemos crear carpetas y cambiar los permisos de carpetas y directorios en el ordenador remoto usando WinSCP. Para crear una carpeta, basta con acceder al menú contextual de la mitad derecha de la ventana en *modo NC* (o bien de la ventana en modo *Explorador de Windows*). Para ello, hacemos clic con el botón derecho del ratón sobre una zona despejada de la ventana, tal como muestra la imagen de abajo:



Alternativamente, podemos utilizar la opción **Nuevo** (**Directorio**) del menú **Archivos**, tras haber activado la mitad derecha (la correspondiente al ordenador remoto) de la ventana de WinSCP (haciendo clic sobre su barra de título). Tras realizar esta acción de cualquiera de las maneras descritas, aparecerá una ventana como la siguiente, pidiéndonos el nombre de la nueva carpeta a crear.



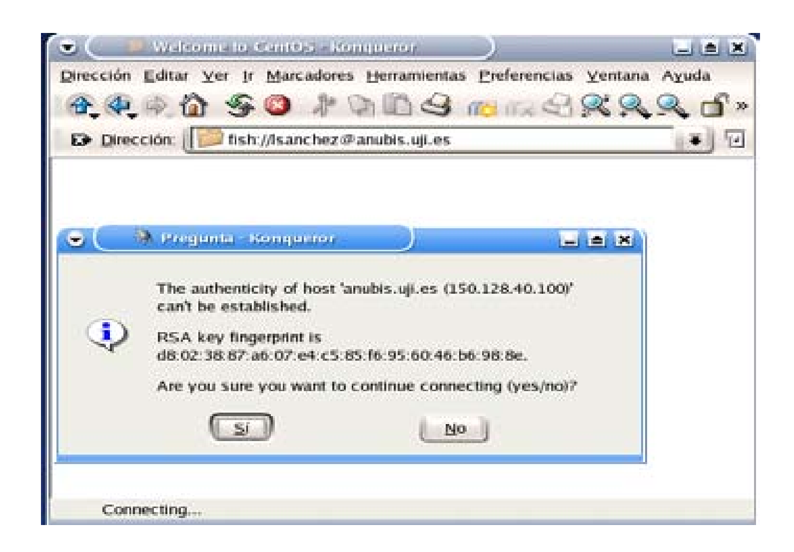
Para cambiar los permisos de un directorio (carpeta) o fichero utilizando WinSCP utilizaremos otro menú contextual, en este caso, el asociado al fichero o directorio cuyos permisos deseamos cambiar. Para activar dicho menú contextual, haremos clic con el botón derecho del ratón sobre el objeto cuyos permisos deseamos cambiar, tal como muestra la imagen siguiente:



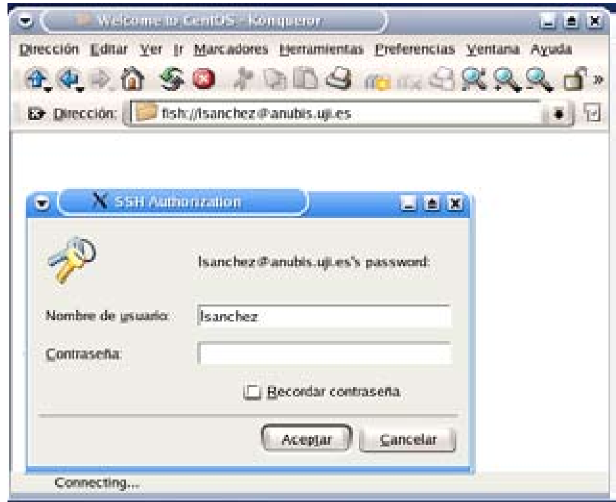
Después accederemos a la opción Propiedades y aparecerá una ventana como la que muestra la imagen de abajo, donde podemos cambiar los permisos a nuestra discreción. El funcionamiento de los permisos en servidores Linux (como es el caso en los ejemplos de conexión mostrados) ya se explicó en el Tema 1 de teoría. Igualmente, puedes consultar el apartado titulado Creación de directorios y permisos de ficheros con Konqueror, más adelante en este documento.

5.2. SCP Y KONQUEROR

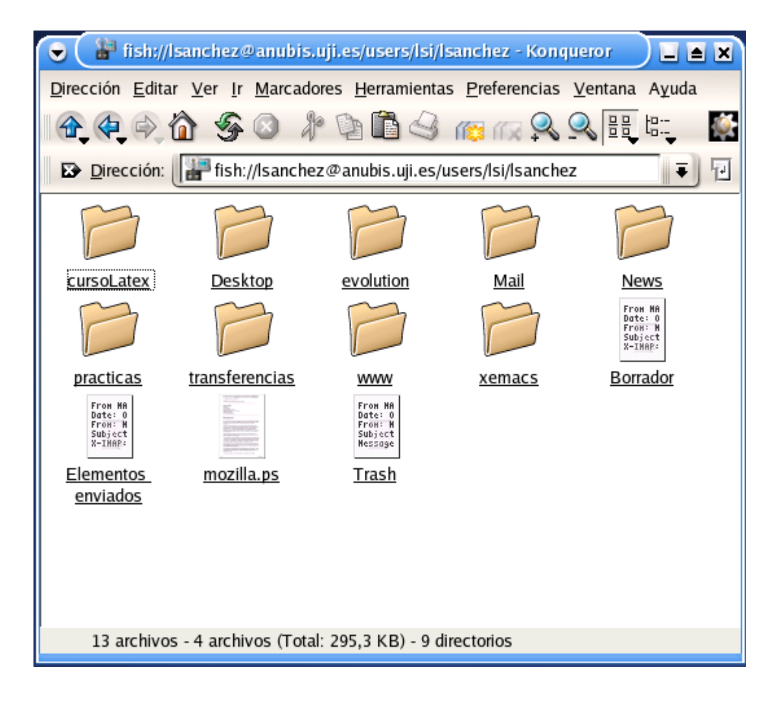
Desde konqueror en linux podemos realizar un SCP en modo gráfico utilizando la URL fish:// en el cuadro de dirección de konqueror. En la siguiente secuencia de imágenes se muestra un ejemplo de conexión de un usuario a su directorio en una máquina a través de SSH.



Paso 1: Solicita confirmación de la firma digital.



Paso 2: Solicita la contraseña.



Paso 3: Muestra el contenido de nuestro directorio.

Una vez la conexión está establecida, el usuario podrá transferir ficheros arrastrando y soltando desde esa ventana al ordenador local y viceversa como si estuviera realizando el trabajo habitual con Konqueror.

Hay que recordar que **fish://** es un mecanismo que únicamente está implementado en Konqueror (no es estándar ni está presente en otros navegadores). Y recuerda que su forma de empleo siempre será (véase la figura del Paso 1 en la página anterior):

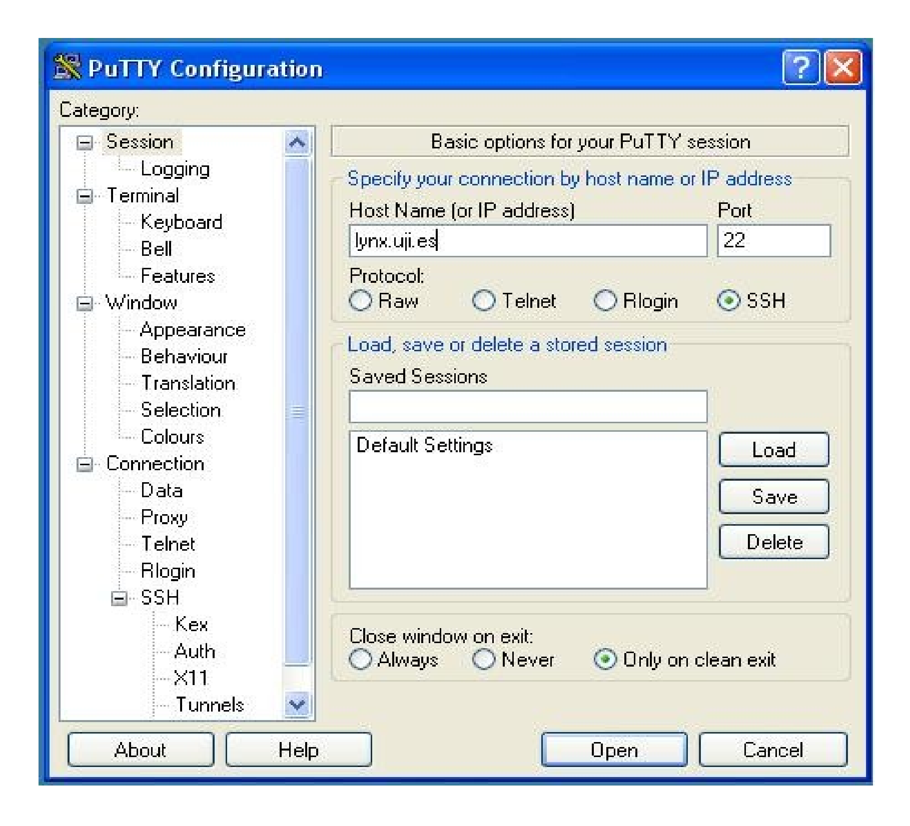
fish://usuario@nombre_servidor

Por otro lado, la creación de carpetas (directorios) y el cambio de permisos de ficheros y/o directorios se hace de la misma manera que se describe en el apartado *Creación de directorios y permisos de ficheros con Konqueror*, más adelante en este documento.

5.3. PUTTY

PuTTY es un cliente Telnet y SSH (Secure Shell) junto con una implementación de un emulador de terminal xterm, que posee multitud de funciones e interesantes opciones. Existen versiones tanto para Windows como para Linux.

Para iniciar una sesión con un ordenador remoto, hay que configurar la conexión proporcionando basicamente el nombre del servidor, el puerto (el 22 para una conexión SSH) y el protocolo, en este caso SSH. Una vez configurado podemos almacenar estos ajustes para utilizarlos en sesiones posteriores, bastará con hacer doble clic sobre el nombre de la sesión para conectarse.



Para establecer la conexión hay que pulsar sobre el botón 'Open' y esperar a que se establezca la conexión y aparezca la pantalla del terminal.

Una vez abierto el terminal, para concluir el proceso de login tendremos que introducir nuestro nombre de usuario y contraseña de nuetra cuenta en el ordenador remoto.

Debes tener en cuenta que, si es la primera vez que nos conectamos al servidor, el programa nos pedirá previamente que aceptemos la firma digital del servidor, tal como hemos visto para el caso de Konqueror y el protocolo **fish://** en el apartado anterior.



Haciendo clic sobre el icono que aparece en la esquina superior izquierda de la ventana accederemos a las opciones del programa. Entre otras opciones podemos cerrar la conexión actual, abrir una nueva sesión o ajustar el tipo de terminal y su apariencia (fuentes, colores,...).

La conexión la podemos cerrar directamente haciendo clic en el botón de cerrar ventana o accediendo a la entrada Cerrar del menú de opciones.

5.4. FTP (FILE TRANSFER PROTOCOL)

Se trata de uno de los servicios más utilizados e interesantes de Internet que, utilizando el protocolo del mismo nombre, nos permite traer/enviar ficheros desde/a un ordenador remoto. Para ello, es necesario:

- Un servidor FTP que nos permita acceder a sus discos y hojear sus directorios para traernos o dejar allí algún fichero.

- Un cliente FTP que pueda interactuar con el servidor utilizando el protocolo del mismo nombre.

5.4.1. Archivadores/compresores.

Es muy frecuente que los servidores FTP tengan sus ficheros en formato comprimido por los siguientes motivos:

- Ahorro de espacio en disco.

- Ahorro de tiempo en la transmisión y menos costes.

Los ficheros comprimidos terminan en ".z", ".gz" o ".tgz" en UNIX. En Windows terminan en ".zip" debido a que este es el nombre del programa más común que realiza la compresión/descompresión.

También existen otros sufijos correspondientes a otros compresores como ".arc", ".lzh", ".arj", ".rar", etc.

Otra posibilidad de los compresores es generar ficheros auto-extraíbles o auto-comprimibles, que incluyen el software necesario para descomprimirse a sí mismos, evitando así que el usuario tenga que poseerlo.

Un empaquetador es un programa capaz de juntar en un único archivo varios ficheros diferentes para facilitar la transmisión, y posteriormente realizar la operación inversa obteniendo los ficheros que vienen juntos. Los compresores actuales son también empaquetadores.

5.4.2. Tipos de ficheros en FTP

Los ficheros ASCII contienen información en texto estándar, y por eso pueden visualizarse en cualquier ordenador después de haber traducido los caracteres de control a los del sistema operativo del ordenador que utilizamos. Esta traducción puede dar a lugar diferentes formas de interpretar un carácter por diferentes sistemas operativos. Son de este tipo los ficheros con extensión .txt, .html, .C y códigos fuente en general.

Los ficheros binarios: sólo pueden interpretarse correctamente en aquel tipo de ordenador o sistemaoperativo para el que estén construidos. Por lo tanto se transmiten bit a bit, sin adaptar nada. Pertenecen a este tipo las extensiones .exe, .bin, imágenes (.gif, .jpg), textos de procesadores, y ficheros de audio y vídeo.

5.4.3. Tipos de conexiones FTP

La **conexión autentificada** exige que el cliente tenga una **cuenta** en el ordenador remoto, con un nombre de usuario y una contraseña. De esta manera están vinculados a cada cuenta los permisos que el cliente tiene sobre los ficheros y directorios. En este caso suele ser posible tanto traer como dejar ficheros, borrar ficheros y directorios, crear directorios, cambiar permisos, etc.

En la **conexión anónima** el cliente entra en una **cuenta** específica para todo

aquel que quiera acceder, y que tiene reducidos al mínimo los permisos sobre la información del servidor remoto: habitualmente sólo es posible hojear el contenido de los directorios y traerse los ficheros que uno desee.

5.4.4. Formas de realizar una conexión FTP

1.- A través del intérprete de órdenes, en modo texto (se verá en la práctica 8).

2.- Utilizando cualquier navegador web, pero especificando el protocolo **ftp://** en lugar de **http://** en la dirección a la que queremos acceder. El navegador también contiene un cliente ftp que nos permite navegar por los directorios del servidor ftp si éste está preparado para ello.

3.- Utilizando un programa específico para FTP en entornos gráficos, como por ejemplo Filezilla (Windows) o gFTP (Linux).

4.- Las descargas de la red que solicitamos con nuestro navegador cuando estamos en una página web, son encomendadas habitualmente a un servidor FTP que se encarga de realizarlas gracias a que nuestro navegador ya contiene, como se ha dicho, un cliente FTP.

5.4.5. Información necesaria para realizar una conexión FTP

- Dirección del servidor.
- Nombre de usuario o login.
- Contraseña para los accesos autentificados.
- Directorio donde se encuentra el fichero que se desea transferir.

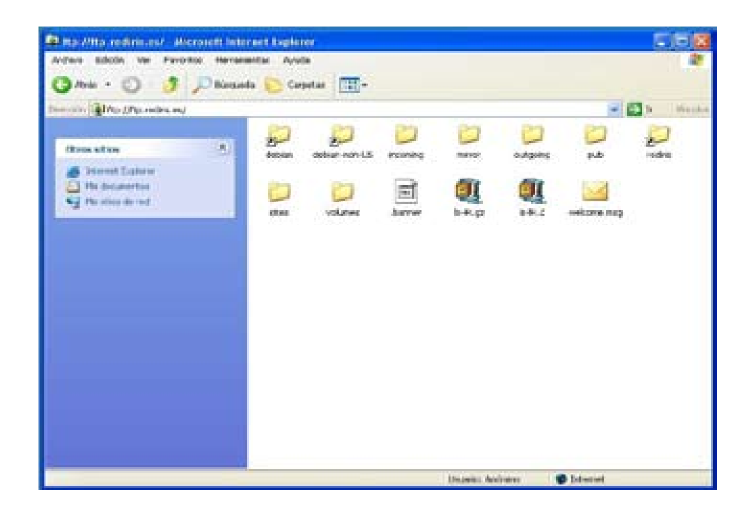
5.4.6. Acceso al servidor FTP a través del navegador

Para tener acceso a un servidor FTP mediante un navegador (Konqueror, IExplorer...) deberemos introducir la dirección del mismo. Para el ejemplo siguiente utilizaremos IExplorer como navegador.

Por ejemplo ftp://ftp.rediris.es

Ten en cuenta que en el caso de usar Internet Explorer hay que arrancarlo en modo *Explorador de Windows*. Para ello, una vez hayas accedido al servidor FTP, se ha de hacer clic en el menú **Ver**, opción **Abrir el sitio FTP en el explorador de Windows**. Otra posibilidad consiste en arrancarlo directamente como *Explorador de Windows*. Abre, por ejemplo, **Mi PC** (doble clic para ejecutarlo) y teclea en la casilla **Dirección:** la dirección del sitio FTP al que deseas acceder, por ejemplo ftp://ftp.rediris.es.

Si el acceso al servidor es anónimo, directamente nos mostrará el listado de ficheros y directorios para poder acceder a los mismos, como se aprecia en la siguiente figura.

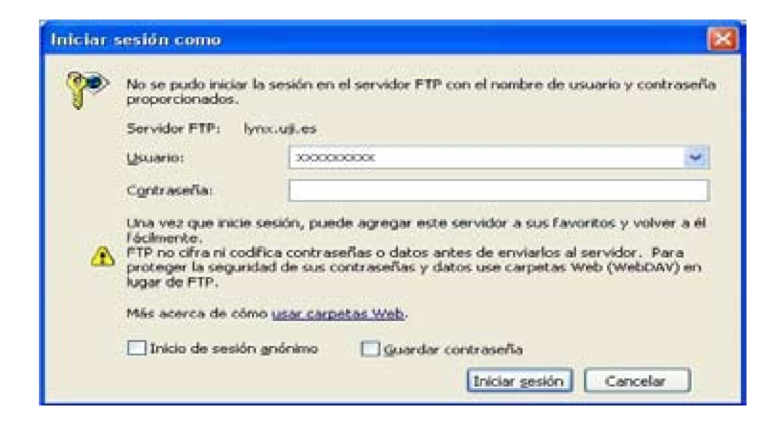


La ventana anterior muestra el contenido del directorio raíz del servidor. En ella encontramos carpetas y ficheros (identificados por el icono correspondiente). Al pinchar sobre cualquier carpeta, accedemos a sus contenidos y podemos, de esta forma, navegar por la estructura de directorios del servidor.

Si el acceso al servidor es autentificado, deberemos introducir nuestro **nombre de usuario** (login) y la **contraseña** (password). Para ello, existen varios métodos en un navegador:

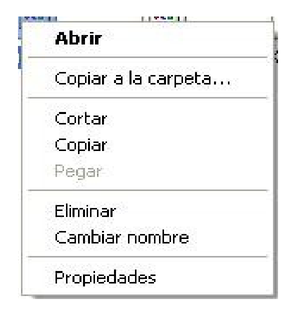
- Introducirlos en la misma dirección del servidor. Por ejemplo ftp://**nombreusuario:contraseña@servidor-ftp**

- Introducir la dirección del servidor, con el usuario solamente (por ejemplo, ftp://xxxxxxxx@lynx.uji.es) y el navegador nos solicitará el nombre de usuario y contraseña, como se aprecia en la siguiente figura.



5.4.7. Transferir archivos

Para transferir los archivos desde el servidor FTP a nuestro ordenador (descargar o *download* en inglés), bastará con arrastrar y soltar donde queramos guardar el fichero, o hacer clic con el botón derecho y aparecerá un menú para indicar qué queremos hacer con el archivo.



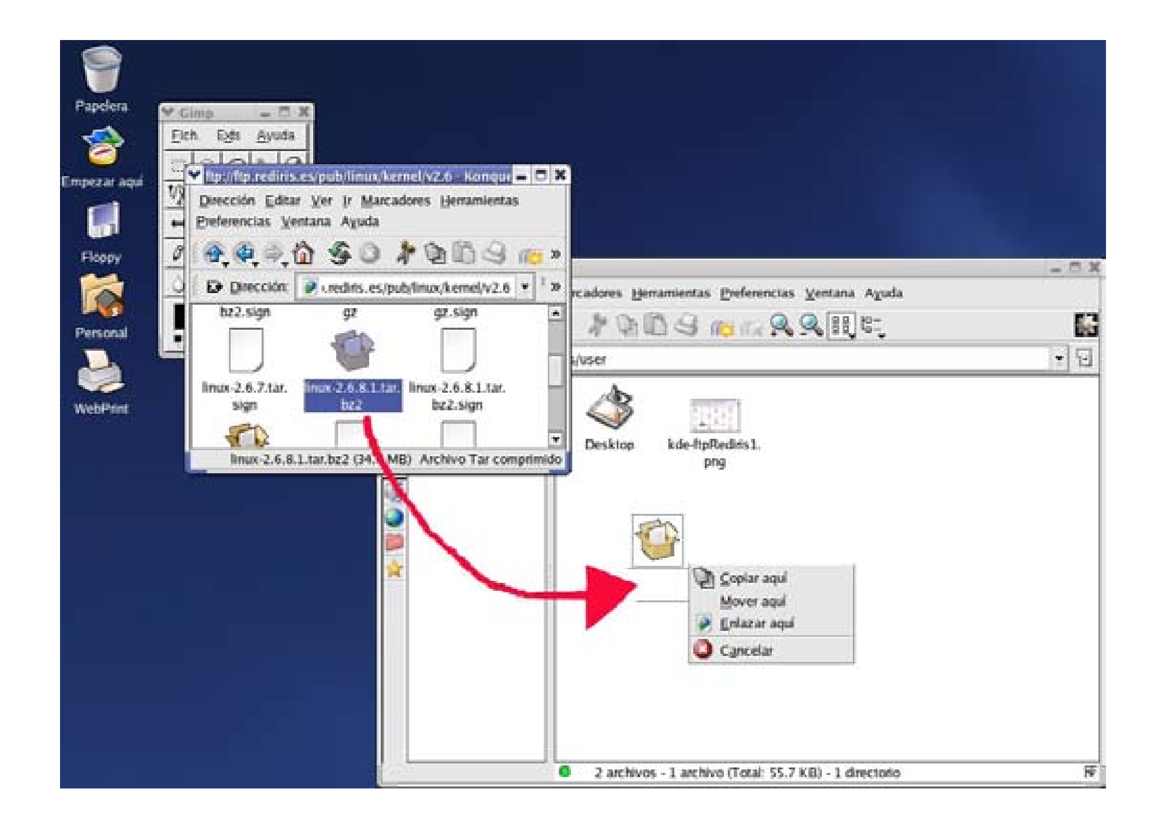
Si elegimos "Copiar a la carpeta", nos preguntará donde queremos guardar el fichero en nuestro sistema de archivos y aparecerá una ventana mostrando el progreso de la descarga. A continuación se muestra la figura como ejemplo:



■ Desde **Iexplorer**: Para enviar un fichero al directorio sobre el que estamos, arrastramos y soltamos donde queramos guardar el fichero.

■ Desde **Konqueror**: La conexión a FTP desde Konqueror está integrada con el sistema de archivos del sistema operativo GNU/Linux, es decir, una vez abierta una ventana con una sesión de FTP, podemos abrir otra y arrastrar los ficheros o carpetas entre ellas como si fueran dos carpetas de la misma máquina. En la siguiente secuencia de imágenes (figuras 5, 6 y 7) se observa la transferencia de archivos desde el servidor anónimo ftp.rediris.es a nuestro directorio personal. Una vez soltemos el ratón (ver la figura que se muestra a continuación) Konqueror solicita que elijamos una opción en el menú que aparece. Deberemos elegir la opción "Copiar aquí".

Utilizando Konqueror, los pasos necesarios para subir archivos al servidor FTP serían los mismos que para realizar una descarga, pero arrastrando desde nuestra ventana local a la ventana en donde hemos abierto una conexión FTP autentificada.

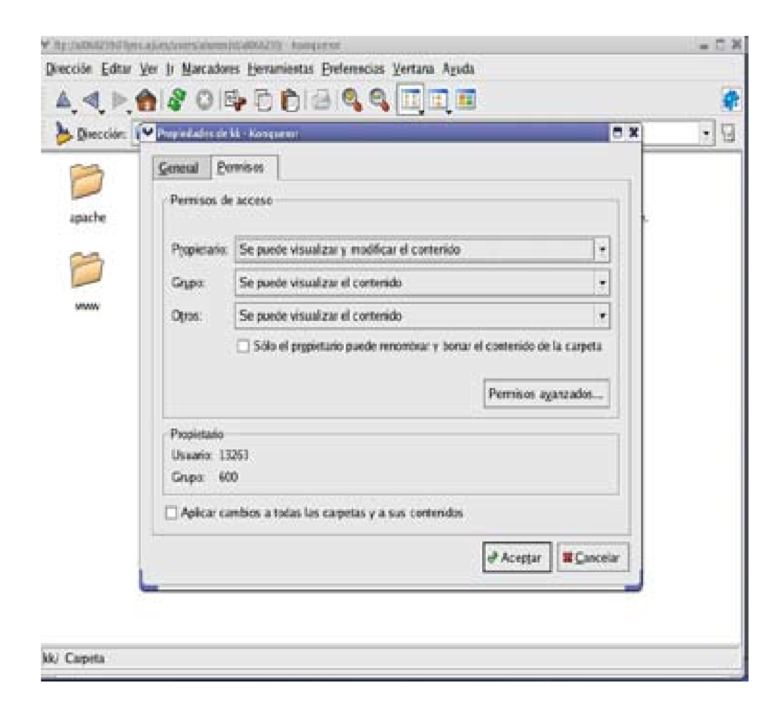


5.4.7.1. Creación de directorios y permisos de ficheros con Konqueror

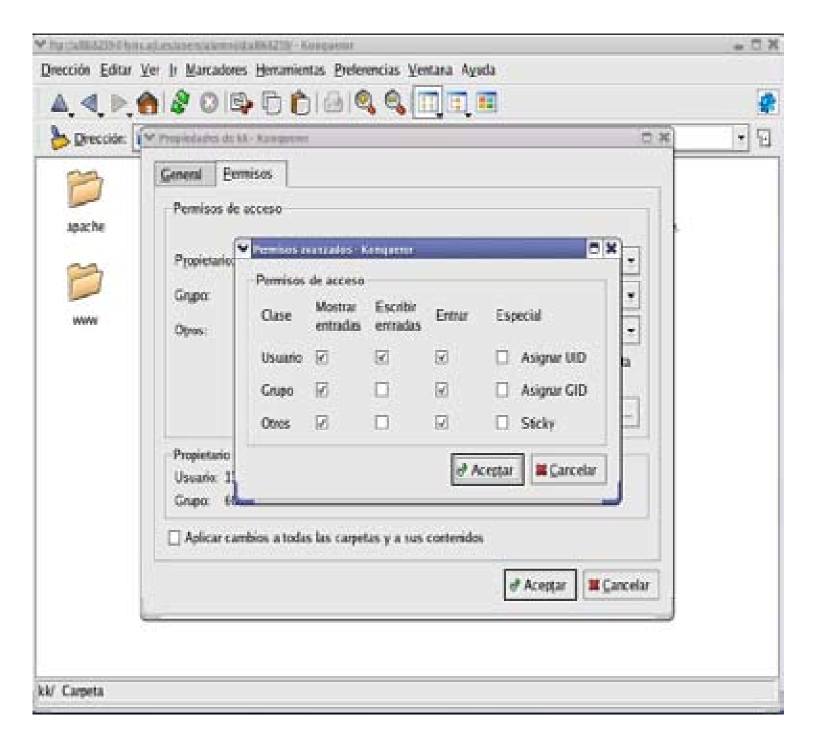
Utilizando Konqueror para conectarnos al servidor FTP, podremos crear directorios así como cambiar los permisos de acceso a los mismos y a los ficheros. Un ejemplo real ocurre cuando ubicamos nuestras páginas web en el servidor para que sean visibles por todo el mundo.

En la figura 8 se observa como crear un directorio una vez hemos realizado la conexión FTP con konqueror. Una forma sencilla de hacerlo es pulsando con el botón derecho del ratón en una zona en blanco de la ventana y eligiendo la opción “crear nuevo” y “carpeta”.

En la página siguiente se presenta la imagen donde se observa una modificación en los permisos de acceso a un directorio en concreto. Esto lo haremos pulsando con el botón derecho del ratón encima de un directorio y eligiendo la opción “propiedades” y luego la pestaña “permisos”.



Si lo que queremos es visualizar los permisos de una forma más amena y en forma de columnas, hay que pulsar con el ratón en el botón “Permisos Avanzados”. La nueva forma de modificar los permisos puede verse en la siguiente figura.

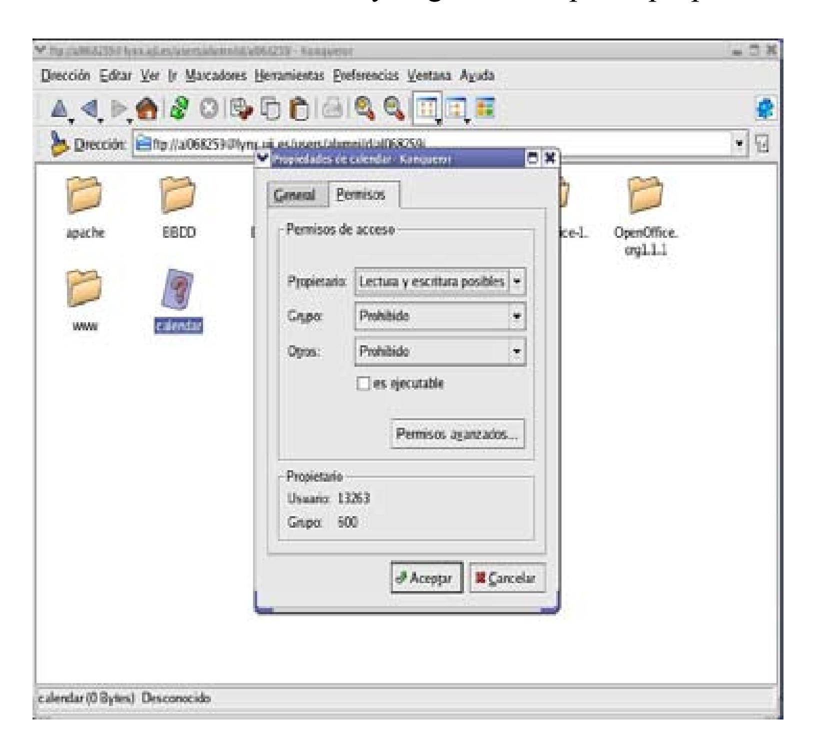


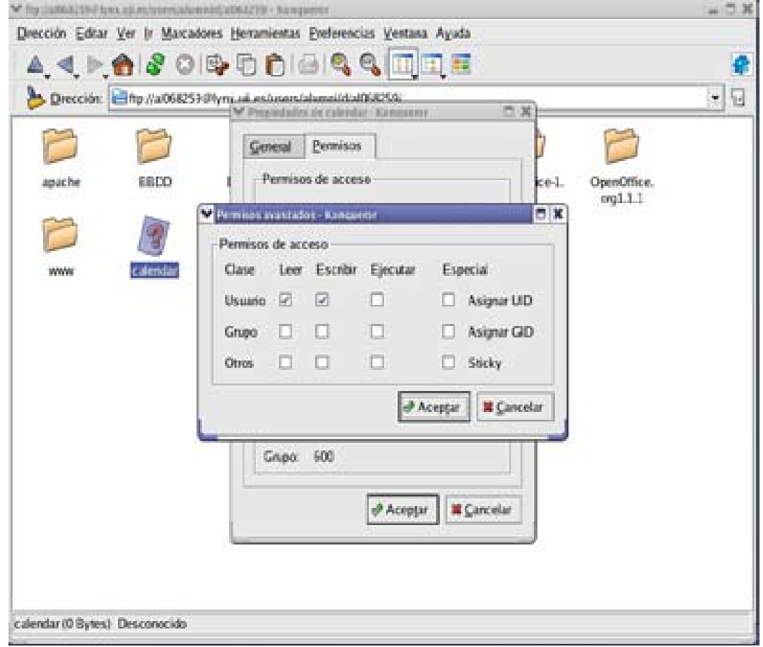
Los permisos no son los mismos para directorios que para ficheros. Ahora explicaremos los permisos de los directorios. Obviamente, un usuario solo puede definir los permisos de un directorio o fichero si es el propietario del mismo, y tiene permisos de escritura sobre el mismo.

Nos centraremos en las columnas “Mostrar entradas”, “Escribir entradas” y “Entrar”. Si queremos que los usuarios puedan ver la información de nuestro

directorio, tendremos que dejar seleccionadas las casillas "Usuario", "Grupo" y "Otros" en la primera y última columna. Si además queremos que los usuarios de nuestro propio grupo puedan añadir archivos o modificar los existentes, deberemos seleccionar la casilla correspondiente de la segunda columna.

A continuación se muestran dos figuras donde se observa una modificación en los permisos de acceso a un fichero en concreto. Esto lo haremos pulsando con el botón derecho del ratón encima de un fichero y eligiendo la opción propiedades.





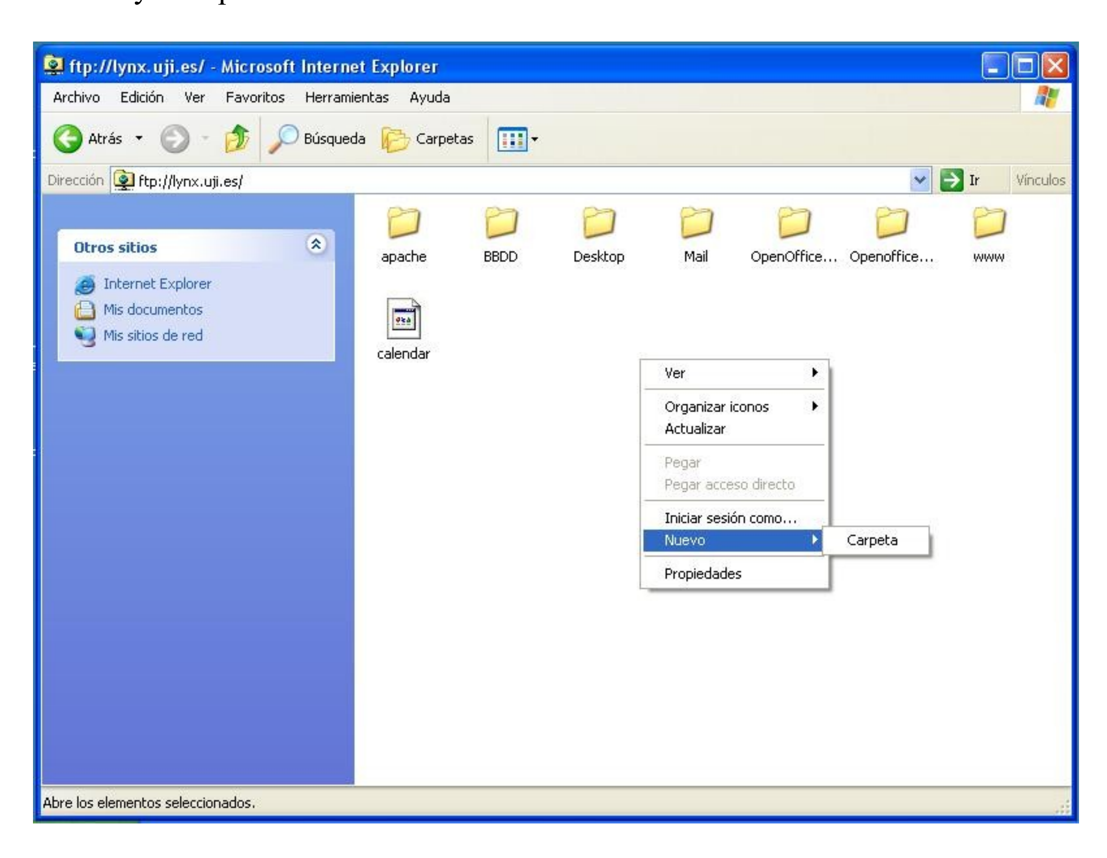
Nos centraremos en las columnas "Leer", "Escribir" y "Ejecutar". Si queremos que los usuarios puedan leer el fichero, tendremos que dejar seleccionadas las

casillas “Usuario”, “Grupo” y “Otros” en la primera columna. Si además queremos que los usuarios de nuestro propio grupo puedan modificar un archivo en concreto, deberemos seleccionar la casilla correspondiente de la segunda columna. La tercera columna indica que el fichero en cuestión es un programa, y activando la casilla correspondiente podremos permitir que cierto usuario lo ejecute.

5.4.7.2. Creación de directorios y permisos de ficheros con IExplorer

Utilizando IExplorer para conectarnos al servidor FTP, podremos crear directorios así como cambiar los permisos de acceso a los mismos y a los ficheros. Un ejemplo real ocurre cuando ubicamos nuestras páginas web en el servidor para que sean visibles por todo el mundo.

En la Figura 13 se observa cómo crear un directorio una vez hemos realizado la conexión FTP con IExplorer. Una forma sencilla de hacerlo es pulsando con el botón derecho del ratón en una zona en blanco de la ventana y eligiendo la opción “Nuevo” y “Carpeta”.



Los permisos en IExplorer son los mismos para directorios que para ficheros. Lo que cambia es el significado que tiene cada uno de estos permisos si es para ficheros o para directorios. Obviamente, un usuario solo puede definir los permisos de un directorio o fichero si es el propietario del mismo, y tiene permisos de escritura sobre el mismo.

En cuanto a los permisos de los directorios, si queremos que los usuarios puedan ver la información de nuestro directorio, tendremos que dejar seleccionadas las casillas “Propietario”, “Grupo” y “Todos los usuarios” en la primera y última columna. Si además queremos que los usuarios de nuestro propio grupo puedan

añadir archivos o modificar los existentes, deberemos seleccionar la casilla correspondiente de la segunda columna.

En cuanto a los permisos de los ficheros, si queremos que los usuarios puedan leer el fichero, tendremos que dejar seleccionadas las casillas “Propietario”, “Grupo” y “Todos los usuarios” en la primera columna. Si además queremos que los usuarios de nuestro propio grupo puedan modificar un archivo en concreto, deberemos seleccionar la casilla correspondientes de la segunda columna. La tercera columna indica que el fichero en cuestión es un programa, y activando la casilla correspondiente podremos permitir que cierto usuario lo ejecute.

CAPÍTULO 6

6. HERRAMIENTAS DE INTERNET DESDE CONSOLA

6.1. INTRODUCCIÓN

En estos apuntes se explica el funcionamiento de una serie de herramientas que permiten conectarse remotamente a otro ordenador para ejecutar órdenes en él y transferir ficheros entre dos ordenadores.

Si aprendes a manejar estas herramientas, podrás conectarse desde tu casa a *lynx.uji.es*, el ordenador en el cual tienes tu cuenta de trabajo en la Universidad, y ejecutar órdenes de forma remota en dicho sistema. También aprenderás a transferir ficheros desde *lynx.uji.es* hasta tu casa y/o viceversa. Recuerda que *lynx* es el servidor "gemelo" de *anubis*, que se usa para efectuar conexiones interactivas (SSH, SCP, FTP, …).

6.2. CONEXIÓN REMOTA

Si un ordenador puede ser utilizado por varios usuarios simultáneamente (normalmente se le denomina *servidor*), es necesario que cada uno de estos usuarios disponga de un terminal propio desde el que acceder al mismo[1]. Es decir, cada uno de los usuarios necesita como mínimo un teclado (y/o ratón) y una pantalla que le permita introducir las órdenes y visualizar la salida de las mismas. El servidor es un *ordenador remoto* (situado a decenas de metros o miles de kilómetros), mientras que el ordenador desde el cual me conecto es el *ordenador local*.

Una vez se ha establecido una conexión entre el ordenador local y el ordenador remoto, el usuario puede ejecutar órdenes en aquél viendo el resultado de la ejecución de éstas en la pantalla del ordenador local. En nuestro caso, nos conectaremos habitualmente a ordenadores remotos con S.O. de la familia UNIX, por lo que las órdenes que podremos ejecutar son las mismas que se han visto para el intérprete de órdenes de Linux: `ls`, `cd`, `mkdir`, `less`, `cp`, `rm`, `mv`, `chmod`, etc.

¿Es posible ejecutar remotamente aplicaciones "gráficas", es decir, que abren ventanas en su funcionamiento? La respuesta es sí, pero antes es necesario configurar ciertas cosas que no comentaremos por estar fuera del alcance de los objetivos de la asignatura. No obstante, debes saber que, en la mayoría de los casos, el administrador de un sistema puede haber preparado algunas de estas "cosas" y que si empleas la opción `-X` en una conexión con `ssh`, podrás ejecutar aplicaciones gráficas remotamente (viendo los gráficos en la pantalla local). Otra posibilidad para ejecutar remotamente programas gráficos es usar un programa cliente (visualizador) que acceda a un servidor VNC (*Virtual Network Computing*): además esta solución es completamente independiente de los Sistemas Operativos y arquitecturas de las máquinas cliente y servidor.

Existen diversas aplicaciones que permiten realizar esta función de conexión remota desde consola. En nuestra asignatura estudiaremos dos de ellas: `telnet` y `ssh`. A estas aplicaciones se les denomina *clientes* ya que permiten acceder, respectivamente, a dos servicios distintos de conexión remota.

[1]No van a estar todos "pegándose" por utilizar el mismo teclado y monitor ;-).

6.2.1. El cliente telnet

La aplicación *telnet* (*terminal emulation*) se puede ejecutar en un ordenador cualquiera y permite acceder de forma remota a otro ordenador. Al iniciar la conexión con el ordenador remoto, éste pedirá al usuario que se identifique introduciendo su nombre de usuario (*login*) y contraseña (*password*).

Cuando ejecutemos `telnet`, deberemos indicarle cuál es el nombre de la máquina (o su dirección IP) con la que deseamos establecer la conexión. Por ejemplo, para conectarse a *lynx.uji.es*, teclearíamos lo siguiente (en un terminal):

telnet lynx.uji.es

Una vez hecho esto, en la parte inferior de la ventana del terminal se muestran dos líneas en las que se informa sobre el Sistema Operativo instalado en la máquina remota y, posiblemente, algunos datos adicionales sobre el sistema. A continuación, en la línea (o líneas) siguientes pueden mostrarse diversos mensajes (según lo haya configurado el administrador). Finalmente, la máquina remota pide al usuario que introduzca su nombre en el sistema (`login:`). Una vez introducido el nombre del usuario, se solicita la contraseña. Los caracteres tecleados en la contraseña no se visualizan de ningún modo, ni siquiera como asteriscos (aunque creas que ha dejado de funcionar, los caracteres que introduces sí se están teniendo en cuenta).

Una vez introducidos tu nombre de usuario y contraseña correctamente, pueden aparecer (según la configuración del servidor por parte del administrador) uno o varios mensajes de aviso informando de las últimas novedades relacionadas con el uso de la máquina remota. Una vez se ha accedido a la cuenta, la máquina remota queda a la espera de recibir órdenes y lo indica mediante el *prompt* del sistema (generalmente de la forma `$`, `usuario@maquina_$` o algo por el estilo). Por ejemplo, podemos teclear la orden `ls -l` para solicitar un listado largo de los contenidos del directorio en el que estamos situados (para empezar, el directorio personal del usuario). Para terminar la sesión de trabajo con la máquina remota, finalizando la ejecución de `telnet`, hay que dar la orden `exit`.

Recuerda que el principal inconveniente de `telnet` es que toda la comunicación con el servidor "viaja" en texto plano (ASCII o alguna de sus extensiones) por la red. Es por ello que alguien malicioso "escuchando" las comunicaciones puede conseguir acceso a información confidencial. Por ello, ha caído en desuso hoy en día, prefiriéndose alternativas mucho más seguras como `ssh`. De hecho, no emplearemos el acceso mediante `telnet` en las prácticas ya que *está deshabilitado en lynx por cuestiones de seguridad*.

6.2.2. El cliente `ssh`

SSH son las siglas de *Secure SHell*. Éste es un servicio de comunicación entre dos ordenadores, uno local y otro remoto, que permite servicios de conexión a distancia como los que facilita `telnet`. Su funcionamiento es exactamente igual

que `telnet`: la única diferencia radica en que toda la información es codificada y cifrada automáticamente antes de ser transmitida a través de Internet (tanto en un sentido como en otro), lo cual permite establecer *conexiones seguras*.

Hay muchos programas basados en SSH. Por ejemplo, uno de los más populares tanto en Windows como en Linux (veáse la práctica anterior) es *PuTTY.* En Linux podemos utilizar la aplicación `ssh` seguida del nombre de la máquina con la que se desea establecer la conexión remota y, posiblemente, indicando el nombre del usuario con el que queremos acceder a la máquina (separado del nombre de la máquina por el carácter `@`). Si no indicamos el nombre de usuario (esto es algo opcional) `ssh` usará el *mismo nombre que tiene el usuario que ejecuta el programa en la máquina local*. Por ejemplo, si el usuario `al000000` en la máquina local quiere conectarse a *lynx.uji.es* en la que tiene el *mismo nombre de usuario*, teclearía:

ssh lynx.uji.es

Si embargo, si quisiera conectarse, por ejemplo, a la máquina `nuvol.uji.es` en la que tiene una cuenta con el nombre de usuario `juancar`, teclearía:

ssh juancar@nuvol.uji.es

En ambos casos, se establece una conexión segura con la máquina remota. No obstante, debes tener en cuenta que si el ordenador local no "conoce" todavía al remoto se mostrará un mensaje de este tipo:

The authenticity of host 'lynx.uji.es (150.128.40.114)' can't be established.
RSA key fingerprint is ee:ab:d4:b6:fe:0d:90:7b:0f:9f:11:41:4d:5f:6b:14.
Are you sure you want to continue connecting (yes/no)?

Si éste es el caso, debes contestar `yes` y pulsar claroEnter. A continuación, el sistema remoto solicita la contraseña. Una vez introducida ésta correctamente, el usuario ya puede ejecutar órdenes en el ordenador remoto (servidor). Para terminar la sesión de trabajo con la máquina remota, finalizando la ejecución de `ssh`, teclearemos la orden `exit` (como con `telnet`).

Si se desean ejecutar aplicaciones gráficas, mostrando la salida (los gráficos) en la pantalla del ordenador local, deberemos emplear la opción `-X`: `ssh -X juancar@nuvol.uji.es` en el ejemplo. Para que esto funcione correctamente, se requieren dos condiciones:

- El administrador del sistema remoto debe haber configurado el servidor `ssh` de manera que admita la posibilidad de usar la opción `-X` en la conexión (en la mayoría de sistemas Linux, por defecto es así al instalar y configurar el servicio `ssh`).

- Tanto el ordenador remoto como el ordenador local deben usar un *sistema gráfico compatible*: el entorno X-Window de Unix. Por tanto, la conexión debe establecerse entre máquinas Unix (recuerda que Linux es una variedad más de Unix).

Además, se aconseja que *la conexión entre ambas máquinas sea rápida* (un ancho de banda grande, en torno a 500 Kbps. o más). Si no, los gráficos se mostrarán demasiado lentamente como para poder trabajar en el ordenador local.

También quisiéramos indicar otra característica interesante de `ssh`. ¿Qué ocurre si solamente queremos ejecutar una orden concreta en la máquina remota y no realmente establecer una conexión? Pues, podemos hacerlo. Imagina que solamente queremos ejecutar la orden `ls -l` en la máquina *nuvol.uji.es* como el usuario `juancar`:

ssh juancar@nuvol.uji.es "ls -l"

La salida del programa sería el resultado de ejecutar la orden `ls -l` en el directorio personal del usuario `juancar` en *nuvol.uji.es*. Si quisiera listar el contenido de un directorio determinado tendría que haber indicado la ruta a ese directorio desde el directorio personal de `juancar`. Observa que la orden a ejecutar se escribe siempre entrecomillada tras el nombre de la máquina remota.

6.3. TRANSFERENCIA DE FICHEROS

FTP (siglas de *File Transfer Protocol*) es el servicio básico que permite la transferencia de ficheros entre distintos ordenadores conectados a Internet. Como la mayor parte de los servicios de Internet, para hacer uso de él se debe ejecutar un programa que permita conectarse desde el ordenador local que estamos utilizando al ordenador remoto que proporciona dicho servicio.

Utilizando este servicio se pueden transferir ficheros a/desde ordenadores en los que tenemos una cuenta, es decir, en los que se nos ha proporcionado un usuario (*login*) y una contraseña (*password*). En nuestro entorno de trabajo estos ordenadores son, por ejemplo, *lynx.uji.es* para los estudiantes y *lepus.uji.es* para el personal de administración y servicios y el profesorado.

Además, también se pueden transferir ficheros entre ordenadores que ofrecen el servicio de *FTP anónimo*. Estos ordenadores suelen utilizarse para *mantener información y software de libre acceso a* todo el mundo: en este caso sólo se pueden realizar transferencias para descargar ficheros desde dichos servidores (por ejemplo, los servidores `ftp.suse.com` o `ftp.rediris.es`*). Otra utilidad que se está dando a este servicio es la posibilidad de entrega remota de documentos como, por ejemplo, la presentación de ponencias a organizadores de congresos científicos para su evaluación: en este otro caso las transferencias sólo se pueden hacer para enviar ficheros.*

En estos apuntes presentaremos algunos de los programas disponibles para Linux. Concretamente, estudiaremos la utilización de los programas de consola `ftp` y `scp`.

6.3.1. *El cliente* `ftp`

Tendrás que ejecutar el programa `ftp` para Linux desde un terminal. Cuando ejecutes este programa, tendrás que pasarle como parámetro el nombre de la máquina (equivalentemente, su dirección IP) a la que deseas conectarte. Por ejemplo, si quieres conectarte al servicio FTP de *lynx.uji.es*:

ftp lynx.uji.es

La máquina remota responderá solicitando el nombre del usuario en el sistema. Una vez introducido el nombre de usuario, se solicita la contraseña. Los caracteres tecleados en la contraseña no se visualizan de ningún modo (aunque no se vean en pantalla, los caracteres que introduces sí se están teniendo en cuenta).

Una vez introducidos tu nombre de usuario y contraseña correctamente, aparecerá, normalmente, un mensaje indicando el S.O. que posee la máquina a la que nos hemos conectado y el modo en que se van a realizar las transferencias: binario o ASCII.

```
$ ftp lynx.uji.es
Connected to lynx.uji.es.
220 (vsFTPd 2.0.1)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (lynx.uji.es:user): al000000
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp>
```

En el modo binario se transfiere un flujo de bits, mientras que en el modo ASCII se transfiere un flujo de bytes, siendo cada uno de ellos un carácter del código ASCII (este modo sólo conviene utilizarlo cuando el fichero que se transfiere contiene caracteres ASCII, como por ejemplo es el caso de los ficheros de texto). Si queremos conectarnos a un FTP anónimo, podemos hacerlo del mismo modo, pero cuando se nos pide el nombre del usuario debemos introducir el del "usuario anónimo", anonymous:

```
$ ftp ftp.uji.es
Connected to lepus.uji.es.
220 (vsFTPd 2.0.1)
530 Please login with USER and PASS.
530 Please login with USER and PASS.
KERBEROS_V4 rejected as an authentication type
Name (ftp.uji.es:user): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Nótese que la máquina remota pide una contraseña para el "usuario anónimo": introduciremos nuestra dirección de correo electrónico como contraseña.

Una vez establecida la conexión `ftp` hay un conjunto de órdenes bastante amplio

que podemos dar. Estas órdenes se muestran al introducir la orden `help`. Las más habituales son las siguientes:

- **`bin:`** establece el modo binario para las transferencias de ficheros que efectuemos con posterioridad.

- **`ascii:`** establece el modo ASCII para las transferencias de ficheros que efectuemos con posterioridad.

- **! orden: `!`** es un carácter "de escape" al intérprete de órdenes *local*. Con él se pueden ejecutar órdenes en el intérprete de la máquina local y ver su resultado en pantalla. Esto puede utilizarse para, por ejemplo, listar el contenido del directorio activo del ordenador local (`! ls -l`) o para crear uno nuevo en dicho ordenador local.

- **`ls:`** lista el contenido del directorio activo en el ordenador remoto. La orden `dir` es equivalente.

- **`cd <directorio>:`** permite cambiar de directorio en el ordenador remoto.

- **`lcd <directorio>:`** permite cambiar de directorio en el ordenador local. Equivale a `! cd <directorio>`.

- **`mkdir <directorio>:`** permite crear un directorio en el ordenador remoto. Para crear un directorio en el ordenador local se usaría `! mkdir <directorio>`.

- **`put <fichero>:`** transfiere un fichero *del ordenador local al ordenador remoto*.

- **`mput <ficheros>:`** transfiere un conjunto de ficheros *del ordenador local al ordenador remoto*. Los ficheros que se transfieren son aquellos que se especifican mediante el uso de los comodines `*` y `?` .

- **`get <fichero>:`** transfiere un fichero *del ordenador remoto al ordenador local*.

- **`mget <ficheros>:`** transfiere un conjunto de ficheros *del ordenador remoto al ordenador local*. Los ficheros que se transfieren son aquellos que se especifican mediante el uso de los comodines `*` y `?` .

- **`prompt:`** cuando se utilizan las órdenes de transferencia `mget` y `mput` existe la posibilidad de que el programa pregunte si realmente se desea o no transferir cada uno de los ficheros especificados (modo interactivo) o bien que, sin solicitar ningún tipo de confirmación, se realice la transferencia de todos los ficheros (modo no interactivo). Pues bien, `prompt off` pone el *modo no interactivo* y `prompt on` activa el *modo interactivo*.

- **close:** cierra la conexión ftp con la máquina remota pero no termina la ejecución del programa, por lo que se puede abrir una nueva conexión con la misma u otra máquina empleando la orden open.

- **bye:** cierra la conexión ftp y finaliza la ejecución del programa.

6.3.2. *El cliente scp*

Éste es un servicio de copia segura: scp (abreviatura de *secure copy*) permite copiar ficheros entre ordenadores distintos a través de la red. Utiliza el mismo modo de identificación que ssh (de la forma usuario@maquina) y proporciona su mismo nivel de seguridad (de hecho, comparten la misma "capa" de encriptación). Por ejemplo, si el usuario al000000 de la máquina local pretende copiar el fichero practica.pdf en su directorio personal de *lynx.uji.es*, ejecutaría (no hace falta especificar usuario en la máquina remota ya que son el mismo):

scp practica.pdf lynx.uji.es:

a lo que *lynx* contestará pidiendo la contraseña del usuario especificado:

$ scp practica.pdf lynx.uji.es:
al000000@lynx.uji.es's password:
practica.pdf   100% |***************************************| 737 00:00
$

Fíjate bien en los dos puntos (:) que aparecen tras el nombre del ordenador remoto en la orden tecleada en el ejemplo.

Una vez introducida correctamente la contraseña, se realiza la copia del fichero al ordenador remoto y termina la ejecución de scp, por lo que vuelve a aparecer el *prompt* del intérprete de órdenes local. Pero si dicho usuario al000000 pretende copiarlo al directorio personal del usuario juancar en *nuvol.uji.es*, ejecutaría[1]:

scp practica.pdf juancar@nuvol.uji.es:

Por supuesto, y al igual que con la orden cp, es posible especificar *rutas*, tanto en origen como en destino:

scp Practicas/IB/practica.pdf juancar@nuvol.uji.es:Asignaturas/IX01/Internet/

Y, por supuesto, *renombrar* la copia efectuada en la máquina remota:

scp Practicas/IB/practica.pdf juancar@nuvol.uji.es:Asignaturas/IX01/Internet/p07.pdf

o incluso emplear *comodines*:

scp Practicas/IB/*.pdf juancar@nuvol.uji.es:Asignaturas/IX01/Internet/

También es posible hacer copias del ordenador remoto al local:

scp juancar@nuvol.uji.es:Pelis/Hulk.avi Pelis/Aver/

Finalmente, quisiéramos indicar que la opción `-r` funciona igual que con la orden `cp`: permite *copiar recursivamente directorios enteros*.

6.3.3. Wget: una herramienta para gestionar descargas

Ésta es una utilidad que sirve para descargar, en general, ficheros de Internet de forma no interactiva. En este sentido, wget no es igual que ftp o scp que pueden funcionar en los dos sentidos. De la misma manera, tampoco permite identificación de usuarios, por lo que se emplea para descargar información pública. Para ello utiliza los dos servicios más utilizados en la red para estos menesteres: http y ftp.

Esta herramienta se ejecuta en modo texto: no es un programa gráfico , ni interactivo, pero tiene gran potencia y flexibilidad. Con wget es posible capturar páginas web (con sus ficheros asociados e incluso directorios ftp) recursivamente, de manera que se pueden hacer copias completas de sitios web que nos interesen. Esta herramienta se encuentra disponible tanto para Windows como para Linux y es software libre.

A continuación, se explica mediante sencillos ejemplos el manejo básico de `wget` (existen un montón de opciones más). Así, para capturar una única página web, por ejemplo `<http://www.uji.es>`, sin los gráficos que la acompañan y sin seguir los enlaces que hay en ella, ejecutaríamos la orden:

wget http://www.uji.es

Si lo que queremos es capturar un sitio web completo, utilizaremos la opción `-m` (`m` indica *mirror*):

wget -m http://www.uji.es

Debemos tener cuidado con esta opción, ya que hará una copia exacta de *todo* el sitio web en nuestro disco duro.

Si tan sólo nos interesa capturar la página principal del sitio web y seguir un determinado número de niveles a partir de los enlaces que hay en dicha página, utilizaremos la opción `-l` (¡ojo! el nivel 0 indica seguir los enlaces que realiza la propia página y nada más). Por ejemplo, para seguir sólo tres niveles en los enlaces utilizaríamos la orden:

wget -r -l3 http://www.uji.es

Con la opción `-r` indicamos que deseamos hacer una captura recursiva y con la opción `-l3` indicamos que se deben seguir tres niveles de recursividad. Hay que tener en cuenta que tal y como estamos ejecutando la orden `wget` se seguirán *todos los enlaces*, tanto los que quedan dentro del mismo dominio como los que llevan a otro dominio distinto (fuera del sitio web original). Para evitar este comportamiento, podemos utilizar la opción `-np` de modo que no se sigan los enlaces que apuntan a

otros sitios web (sólo se accedería a las páginas web de un directorio siguiendo los enlaces que llevan a páginas y/o ficheros que están en sus subdirectorios):

wget -r -l3 -np http://www.uji.es

La opción `-k` permite convertir los enlaces no relativos en enlaces relativos localmente de modo que los enlaces también funcionen en el sitio web local que se está copiando.

La información copiada mantiene la misma estructura que tenía en el servidor y se copia en un directorio que se crea en el ordenador local (en el directorio activo donde estamos situados) con el mismo nombre del sitio web al que accedemos para copiar[1]. En el caso de nuestro ejemplo el directorio se llamará `www.uji.es`.

Finalmente, `wget` también puede resultar de gran utilidad cuando queremos descargar un fichero de gran tamaño al que podemos acceder públicamente. Al tratarse de una descarga que puede tardar bastante tiempo (incluso horas) podría interrumpirse en algún momento, obligando a recomenzar la descarga cuando nos diésemos cuenta de la circunstancia. Si usamos la opción `-c` evitamos tener que comenzar a descargar el fichero desde el principio: si en el directorio donde estamos situados existe una copia *parcial* del fichero, la descarga proseguirá desde el punto en el que se interrumpió:

wget -c ftp://pirates.movies.org/Movies/Hulk.avi

Observa que, en el anterior ejemplo, estamos accediendo a un *servicio público de FTP* (FTP anónimo). Y es que wget "entiende" igualmente el protocolo FTP, de la misma manera que el HTTP y el HTTPS (como los navegadores). Además, `wget` permite utilizar *comodines* para especificar múltiples

ficheros a descargar. Por ejemplo, si quisiéramos descargar todos los ficheros `AVI` del sitio anterior, ejecutaríamos:

wget -c ftp://pirates.movies.org/Movies/*.avi

Para más información, puedes consultar la ayuda del propio programa con la orden `wget --help | less` o bien ejecutar la orden `man wget`.